MUSIC VIDEO AUGMENTED REALITY FOR IPHONE

JAMIKORN AROUNRAK

VORAVAN CHARNSAWAT

A SPECIAL PROJECT SUMMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR DEGREE OF

BACHELOR OF SCIENCE

IN COMPUTER SCIENCE (INTERNATIONAL PROGRAM)

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACEDEMIC YEAR 2011

Thesis Title	Music Video Augmented Reality for IPhone
Student	Mr. Jamikorn Arounrak
	Ms. Voravan Charnsawat
Degree	Bachelor of Science
Program	Computer Science (International Program)
Year	2011
Thesis Advisor	Asst. Prof. Dr. Korakot Prachumrak

ABSTRACT

This special project emphasizes on developing the AR technology. The AR application tracking marker is used to define the scale and coordinate system. The 3D models of the singers will appear on the Iphone screen. The user can choose the music and interact with our application by choosing the singers, cloths and particle effects. Our application can simulate the singer's 3D model in the real time and in the real world for an entertainment.

ACKNOWLEDGEMENT

We would like to express our sincere thanks to our thesis advisor, Asst. Prof. Dr. Korakot Prachumrak for her invaluable advice and be guidance of this thesis since start until successful. We would like to thank our committee, Assoc.Prof.Dr.Jeeraporn Werapun and Dr.Warangkhana Kimpan for their encouragement and insightful comments.

We would like to special thank Mr.Chatpong Suteesuksataporn, Executive Producer of ProGaming institute for his valuable advice and friendly help in Unity engine.

We warmly thanks James Alliban, Emma Schweitzer, and String™ Augmented Reality support team for their guidance and advice in Augmented Reality technology.

We most grateful to Mr. Zac Jingjit, grammar teacher of Easy English School for advice the English grammar for us.

Finally, we would like to thank our family, our friends for friends for help and encouragement.

Jamikorn Arunrak

Voravan Charnsawat

Table of contents

Table of contents (Con't)

Table of contents (Con't)

Table of contents (Con't)

Table of contents (Con't)

List of Tables

Page

List of Figures

List of Figures (Con't)

List of Figures (Con't)

List of Figures (Con't)

List of Figures (Con't)

List of Figures (Con't)

List of Figures (Con't)

List of Figures (Con't)

List of Figures (Con't)

List of Figures (Con't)

List of Figures (Con't)

List of Figures (Con't)

List of Figures (Con't)

List of Figures (Con't)

List of Figures (Con't)

List of Figures (Con't)

Chapter 1

Introduction

1.1 Rational and Research Motivation

Nowadays, smart phone is an important factor for human's lives, especially Iphone that became famous over the last few years. This special project aim to create a new innovation between AR application and the music for the Iphone for the users. Then the users can enjoy visual and audio simultaneously.

Augmented Reality (AR) has become one of the most popular topics in academic research fields. Also it is the state of the art interface design of innovative mobile learning application system development. Augmented Reality allows the user to see the real world in real time with virtual objects overlaid upon. It is composited with the real world to increase user awareness. Augmented Reality technology has many possible applications across a wide range of fields, including entertainment, education, medicine, military training, engineering, and manufacturing.

Therefore, this special project emphasizes on developing the AR technology. The AR application tracking marker is used to define the scale and coordinate system. The 3D models of the singers will appear on the Iphone screen. The user can choose the music and interact with our application by choosing the singers, cloths and particle effects. Our application can simulate the singer's 3D model in the real time and in the real world for an entertainment.

1.2 Objective

The objective of this special project is for the user to see 3D augmented reality singer models on the Iphone screen, while listen to the music. It is done by using String™ library software for MacOS.

1.3 Scope

1. To develop an application on Iphone for tracking marker and link it to music.
2. To develop an application on Iphone that can calculate based on data marker tracking and transform to 3D singer models.
3. To expand the entertainment for example users can interact with model by choosing the singers, cloths and particle effects.

1.4 Organization

This special project consists of 5 chapters.

Chapter 1: Introduction
Chapter 2: Background
Chapter 3: Design and Implementation
Chapter 4: Implementation
Chapter 5: Conclusion

1.5 Stage of special project

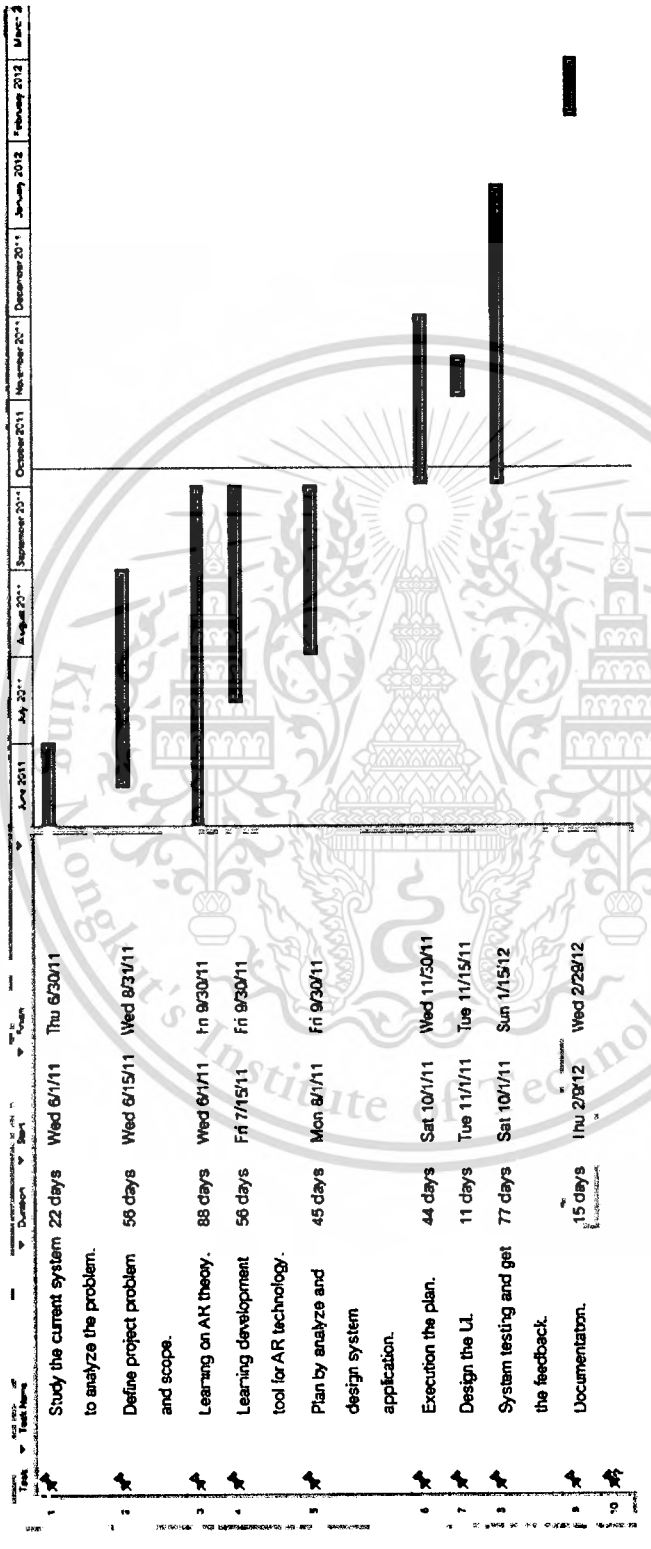


Figure 1.1: Schedule of this special project.

Chapter 2
Background

2.1 Augmented Reality

Virtual Reality is the evolution of technology that begins from research and development for military and flight simulation in United State of America since 1960-1969. An assortment of virtual reality based on the user interface as follows.

a) Desktop VR or Window on World System (WoW) is a virtual reality system use on computer monitor to represent display.
b) Video Mapping as a video device or input data from user and use computer graphics to represent in two-dimensional or three-dimensional model and the user will see themselves and can change themselves from the screen.
c) Immersive Systems is a virtual reality system for personal users by the user use Head mounted displays (HMD).
d) Telepresence is virtual reality system use the remote sensing devices are installed on the robot for connected to the user.

Augmented Reality (AR) is a type of reality technology mixing base on real world with virtual world for the user can have interaction in real time at the same place. This technology is the innovation since 2004. Augmented Reality is a field of research in computer science with the addition of a virtual three-dimensional computer generated models to the images captured by video cameras, web cam or camera of mobile phone by computer graphic technique. Nowadays, Augmented Reality technology was combine to imaging software and other devices and bring on monitor screen or device screen, allow the user was use virtual reality technology with working online can have interact in real time between user and devices of three dimension model that have 360 degree view port by user do not need to go to real place.

Milgram (1994) introduced a taxonomy that relates augmented reality to virtual reality as different degrees of reality-virtuality continuum. Reality-virtuality continuum is shown in Figure 2.1 In the left end of the reality-virtuality continuum is the real environment. A completely immersive virtual environment is in the other end. Augmented reality is near the real environment

end, as it consists of some synthetic elements that overlap the actual real environment. The inverse case where real world content contributes to synthetic surroundings would be called augmented virtuality (Milgram and Kishino, 1994) Milgram also considers augmented reality and augmented virtuality as different levels of the broader concept of mixed reality, even though the term augmented reality has become quite popular in literature.

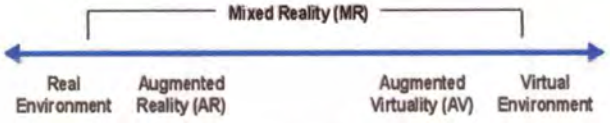


Figure 2.1: Simplified representation of Milgram's Reality-Virtuality continuum after Milgram and Kishino 1994. [1]

2.2 Augmented Reality Application domains

AR is rapidly gaining popularity, but it is still a fledgling technology. Most of the current applications are academic, but there are some commercial products on the market. These applications include everything from factory floor design to storytelling. Based on the wide breadth of these uses, it's clear that AR has a promising future. The next section highlights some examples of contemporary AR projects.

2.2.1 Augmented Reality application in medical

Medical applications are evident. Surgeries are a prime example of use for augmented reality. For instance, surgeries can be performed according to pre-gathered information, which is imposed over the view of the subject. Information could include volume rendered images of physical implants, tumors, or other details, as well as information about the procedure and state of the patient. This way a surgeon can take full advantage of x-rays and other knowledge about the patient on the spot.

Figure 2.2: Experiments with the medical augmented reality system. [2]

2.2.2 Augmented Reality in entertainment

A simple form of augmented reality has been in use in the entertainment and news business for quite some time. When you watch an evening weather report the weather reporter often stands in front of changing weather maps. In the studio the reporter is actually standing in front of a blue or green screen. This real image is augmented with computer generated maps using a technique called chroma-keying. It is also possible to create a virtual studio environment so that the actors appear to be positioned in a studio with computer generated decorating.

(a) (b)

Figure 2.3: (a) live action, (b) combined video. [3]

2.2.3. Augmented Reality for Military Training

The military has been using displays in cockpits that present information to the pilot on the windshield of the cockpit or the visor of their flight helmet. This is a form of augmented reality display. SIMNET, a distributed war games simulation system, is also embracing augmented reality technology.

Figure 2.4: AR application in military. [4]

2.2.4. Augmented Reality for Engineering Design

Imagine that a group of designers is working on the model of a complex device for their clients. The designers and clients want to do a joint design review even though they are physically separated. If each of them had a conference room that was equipped with an augmented reality display this could be accomplished. The physical prototype that the designers have mocked up is imaged and displayed in the client's conference room in 3D.

Figure 2.5: Engineering design using an augmented reality display. [5]

2.2.5. Augmented Reality for Architecture

Architecture is another field that could benefit from AR. The benefits could be not only in displaying on-site information about a new design but also by displaying information about the building, its maintenance or repairing of sites. Using Mobile Augmented Reality (MAR), an architect can receive real time data about the progress of the

construction of a building or other information required Helping the interior design architect is another possible application for the MARS. The system could help the designer to visualize the setting of furniture and share the view with colleagues at a remote office.

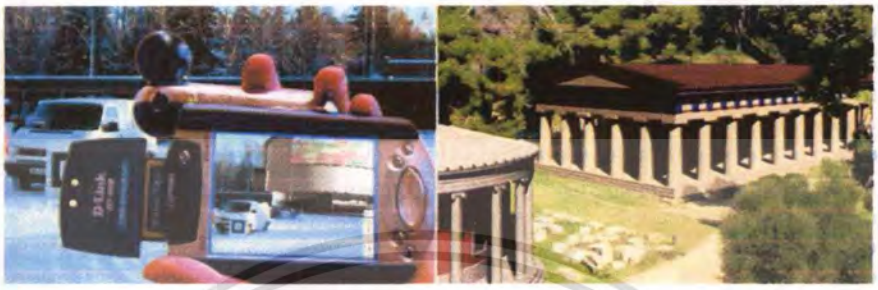


Figure 2.6: Left-Virtual building in PDA. Right-Virtual Hera temple in historical. [6]

2.2.6 Augmented Reality for Business

The market of mobile phones initially targeted the selling of devices to the business community, doing the same for MAR devices could also make it acceptable quicker. Modern business people require more information access from various places and personal information management combined with a MARS could fulfill their needs of accessing information anytime, anywhere. There are already some companies that offer specialized solutions or kits for AR. The services and applications offered for a business person could vary from browsing the stock exchange to accessing a virtual office and sharing documents on the fly. Such an infrastructure could also provide for collaborative work and the access to virtual spaces in which multidisciplinary teams would be able to work over VR models. Using an AR interface that extends the view, is mobile and can handle more interaction, would help the user to achieve faster results and at any time that the work requires.

Figure 2.7: The AR application of Tissot watch. [8]

2.3 Marker base Augmented Reality

When the developer makes the marker base on Augmented Reality, the border of the marker must be square, after system detected the marker it will find the coordinate and position of marker to display 3D models.

The marker including 2 part:

a) Boundary: Use for detect the square area of marker and separate the marker from physical background.

b) Picture inside boundary: Use for identify which 3D models will be display on that marker.

Constrain of the marker:

a) The marker must be square.

b) Have to continue border and color must different from background color (usually use black and white color). The thickness of border must be 25% of edge of length of marker.

c) Area inside boundary marker used for identify the marker and in state of picture that must not be symmetric rotational. The color can be black or white or another color.

Table 2.1: Size of picture inside the marker that effected to camera distance and the marker.

Pattern Size (inches)	**Usable Range** (inches)
2.75	16
3.50	25
4.25	34
7.37	50

If developer creates complex picture, it will reduce the marker distance when the application detect on the marker such as picture size 4.25 inches, user can stay far from camera only 15 inches if create complex picture.

Marker pattern file is the file that collected picture inside of boundary marker, then find the marker's position for find picture from Video frame and separate each marker.

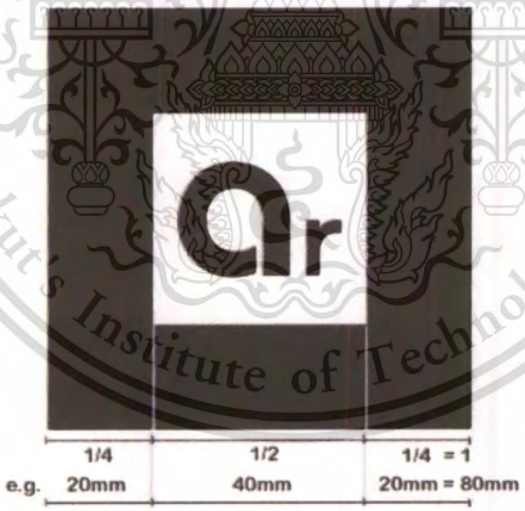


Figure 2.8: Complete marker.

Step of the marker base on Augmented Reality including:

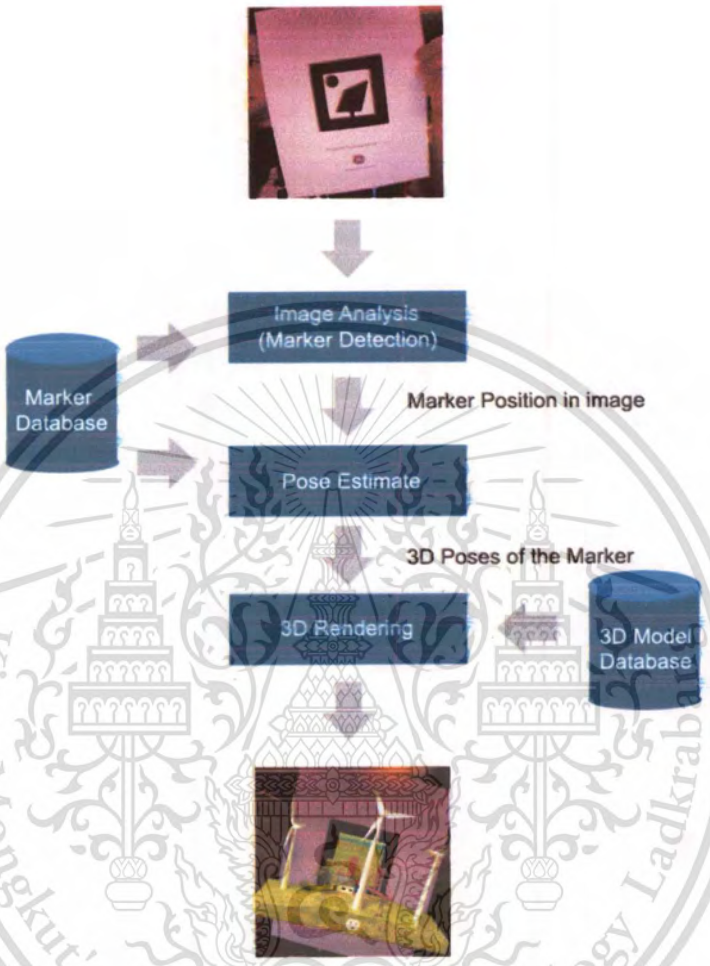


Figure 2.9: Step of the marker base on Augmented Reality.

a) Image Analysis
b) Pose Estimation
c) 3D Graphic Rendering

- Image Analysis / Marker Detection

This strategy is to find marker from picture that detect from camera. First, developer has to collect important data for marker in to database including size of the markers (in centimeter) and format of the marker. Generally, the format of marker must be square, black boundary, white background and area inside boundary marker is black.

The process of image analysis will give the data of markers that user want. When the system search the marker from camera frame can describe from Figure 2.9

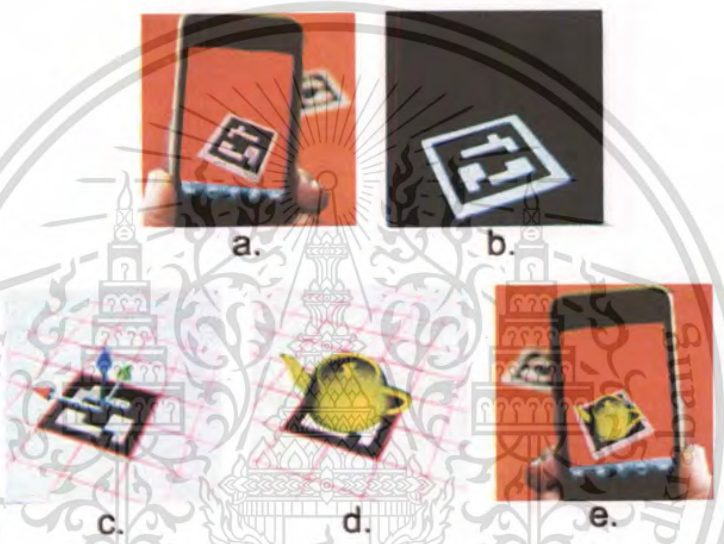


Figure 2.10: Process of Image Analysis.

a.) Use camera to capture picture.

b.) Transform the picture to Binary image by identify each pixel to be 0 or 1. If the binary number is 0, it means intensity of that pixel is lower than threshold and otherwise the boundary number is 1.

c.) Find the connected components by use the connected component labeling technique.

d.) The system will find contours of connected components area.

e.) The system will estimate parameter of linear equation that substitutes all of 4 sides of perimeter square. Then, system will find the corners point from 4 crossing lines and send to pose estimate process.

- Pose Estimation is procedure that calculate 3D Pose of marker, the result will represent in $4\times4(T_{CM})$. There is identifying relationship between camera Coordinated Frame and Marker Coordinated Frame. Show in equation 2.3.1.

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_M \\ Y_M \\ Z_M \\ 1 \end{bmatrix} = \mathbf{T_{CM}} \begin{bmatrix} X_M \\ Y_M \\ Z_M \\ 1 \end{bmatrix} \quad (2.1)$$

Camera coordinated frame or coordinated frame use for refer the camera's position. Marker Coordinated Frame is Coordinate Frame that use for refer marker's position.

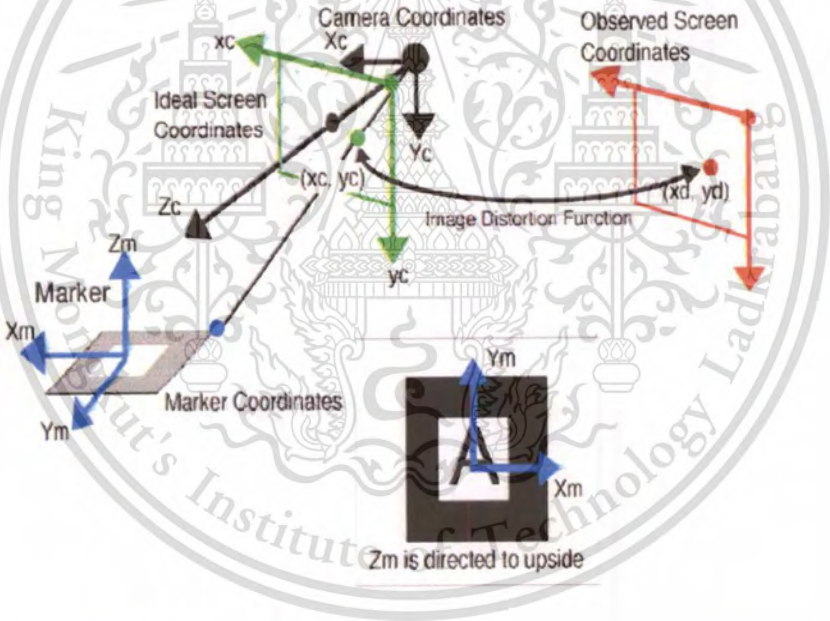


Figure 2.11: The relationship between Camera Coordinated Frame and Marker Coordinated Frame. [9]

From Figure 2.11 show the relationship between point (X_c,Y_c,Z_c) on Camera Coordinated Frame and point (x_1,y_1) in Ideal Screen Coordinated Frame that according to Perspective Projection in equation 2.2

$$\begin{bmatrix} hx_I \\ hy_I \\ h \end{bmatrix} = \begin{bmatrix} sf_x & 0 & x_c & 0 \\ 0 & sf_y & y_c & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = C \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \qquad (2.2)$$

Where C, which is the matrix 3 4 that consist of s,f_x,f_y,x_c,y_c. The fifth value call Camera Parameters that calculate from Camera Calibration procedure and the relationship between point on Ideal Screen Coordinated Frame (x_1,y_1) with Observe Screen Coordinated Frame (x_0,y_0), which is the real point that user see in real picture. Show in Figure 2.11

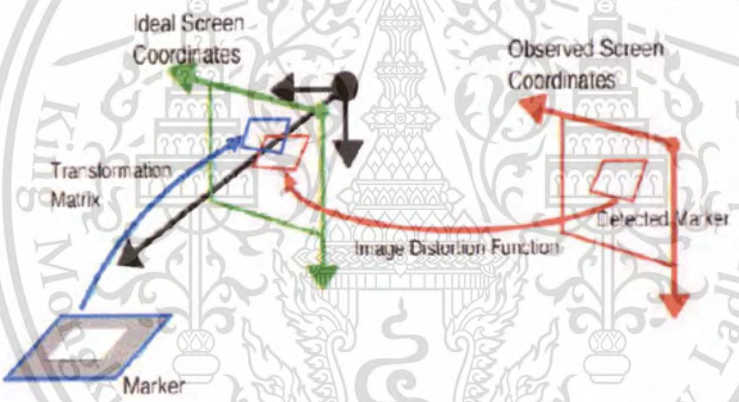


Figure 2.12: Represent the relationship between Ideal Screen Coordinates and Observe Screen Coordinates. [10]

Show in equation

$$\begin{aligned} d^2 &= (x_1 - x_0)^2 - (y_1 - y_0)^2 \\ p &= 1 - fd^2 \\ x_0 &= p(x_1 - x_0) + x_0 \\ y_0 &= p(y_1 - y_0) + y_0 \end{aligned} \qquad (2.3)$$

x_0, y_0 is Center Coordinates of Distortion and f is Distortion Factor. Both values are calculated from Camera Calibration process.

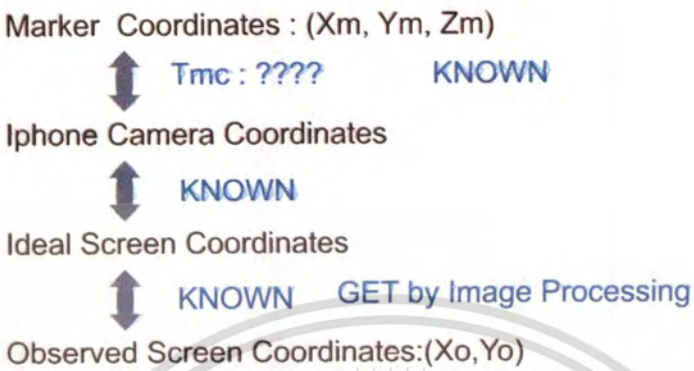


Figure 2.13: Show the process to calculate the 3D Process.

From Figure 2.13 show process to calculate T_{CM}. When the user know 4 points of marker's position on observed Screen coordinates that captured from camera. This value can calculate from Error Function, shows in equation 2.4 Generally, T_{CM} can calculate from Optimization process, which is Iterative process.

$$err = \frac{1}{4}\sum_{i=1}^{4}\{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2\} \quad (2.4)$$

It can show as:

$$\begin{bmatrix} h\hat{x}_i \\ h\hat{y}_i \\ h \end{bmatrix} = C \cdot T_{CM} \begin{bmatrix} X_{mi} \\ Y_{mi} \\ Z_{mi} \\ 1 \end{bmatrix}, i = 1,2,3,4 \quad (2.5)$$

- 3D rendering.

This is the last process to complete Augmented Reality, which is augment the 3D model in to picture that capture from camera at the position of marker.

2.4 Introduction to String Augmented Reality library

On 16th June, 2011, String™ officially launched the fastest, most powerful augmented reality technology for iOS, allowing user to create incredible, cutting-edge AR experiences.

The String AR is fast and flexible with powers 3D animation, games,video, audio and internet based content. String AR recognises framed images and understands where they are in 3D space. It's can plug in into any iOS project, that lets the user display 3D graphics on top of the camera view as if they existed in the real world.

The user can create Augmented Reality apps more by integration with Unity, which is a game development tool for multi-platform game creation include web player, PC and Mac Standalone, IOS, Android, Xbox360, PS3, Wii. The user can literally get started without a single line of code.

- Clean and simple tutorial project to get start.
- In-editor preview plug-in for instant testing.
- A single static library, no header files needed.
- Add maps, messaging, location tracking and other iOS goodness to your AR apps.

2.5 Introduction to Autodesk Maya

Maya was originally a next-generation animation product under development at Alias Research, Inc. based on code fromThe Advanced Visualizer, PowerAnimator and Alias Sketch. The code was ported to IRIX and animation features were added. The codename for this porting project was Maya. Walt Disney Feature Animation collaborated with the maya

development during its production of Dinosaur movie. Disney requested that the User interface of the application be customizable so that a personalized workflow could be created. This was a particular influence in the open architecture of Maya, and partly responsible for it becoming so popular in the industry.

After Silicon Graphics Inc. acquired both Alias and Wavefront Technologies, Inc., Wavefront's next-generation technology was merged into Maya. SGI's acquisition was a response to Microsoft Corporation acquiring Softimage, Co.. The new wholly owned subsidiary was named "Alias|Wavefront".

In the early days of development, Maya started with Tcl as the scripting language, in order to leverage its similarity to a Unix shell language. But after the merger with Wavefront Sophia, the scripting language in Wavefront's Dynamation, was chosen as the basis of MEL (Maya embedded language).

Maya 1.0 was released in February 1998. Following a series of acquisitions, Maya was bought by Autodesk in 2005. Under the name of the new parent company, Maya was renamed Autodesk Maya. However, the name "Maya" continues to be the dominant name used for the product.

Autodesk Maya software is provides a number of tools for creating complex characters and animations. Maya's powerful feature set gives you an almost unlimited power to create any kind of animation.

The functionality of the Maya software can be extended with the use of MEL (Maya embedded language). MEL can be used to customize the user interface and write scripts and macros. The Maya allow user to create objects, lights, cameras and textures. Any object, light, camera, or just any entity can be animated by changing the value of its parameters in time. Maya can be used to achieve more complex effects and animation is as compared to other software is on the shelf. This tutorial is aimed at teaching a newbie, the basic concepts and functionality of the Maya software.

Maya addresses the challenges faced by everyone from artists handling an entire project alone or with a small team to chief technology officers managing a complex production pipeline. From previsualizing feature films before a single frame is shot to producing spectacular, Oscar®-winning finished effects or stereoscopic projects, Autodesk Maya helps studios and artists increase the quality and efficiency of their film pipeline. Autodesk Maya software allows you to

- Access a large pool of trained and talented artists
- Create 3D animatics for previsualization of a setting
- Model, animate, and light realistic high-resolution characters, environments, and performances
- Work interactively with visual media, regardless of bit-depth or image size, using the Maya Composite compositor.
- Extract accurate 3D camera and motion data from video and film sequences and then insert CG elements seamlessly into a scene using Autodesk MatchMove software.
- Roll out complex and efficient pipelines because the software is readily extensible, collaborative, and highly compatible with other toolsets, such as Autodesk MotionBuilder, Autodesk Flame, and Autodesk Inferno.

2.6 Introduction to Xcode 4.1

Xcode is a suite of tools. Xcode is completely support objective-C, C and C++. Xcode developed by Apple, use for developing software for Mac OS X and iOS. Xcode 4.1, is available on the Mac App Store for free for Mac OS X 10.7, and on the Apple Developer Connection website for free to registered developers for Mac OS X 10.6. As of September 2011. Mac OS X install DVDs typically include a copy of Xcode, as well.

Xcode 4.1 coincides with the release of Lion and includes many improvements, some to help adopt new technologies in Lion and some just to make Xcode a better IDE. Lion introduced the concept of full screen apps, and Xcode 4.1 is proper full screen citizen. The user interface of Xcode 4.1 are show in Figure 2.15

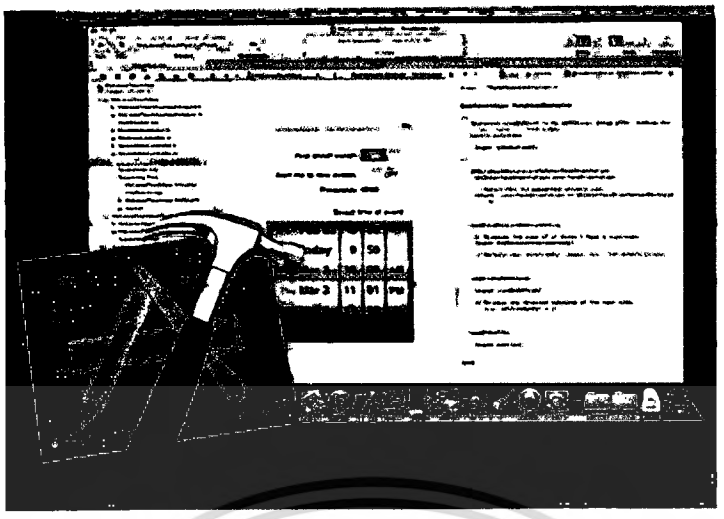


Figure 2.14: Xcode 4.1

2.6.1. Single Window

Xcode 4 is that the many windows used to perform the development tasks developer work on every day have been consolidated into a single window. The Xcode 4 work area has several unique UI elements that make it easy to work on many different tasks, even multiple projects, without cluttering work area. Developer editor is always front and center.

Figure 2.15: Single Window.

2.6.2. Navigators

On the left side is a collection of navigators, including the list of files in your project, sorted symbols, a central search interface, issue tracking, debugging data with compressible stack traces, active and inactive breakpoints, and a persistent collection of logs. The unified navigator UI provides live filtering of content and search results, so developer can focus on current task.

Figure 2.16: Navigators.

2.6.3. Jump Bar

At the top of every editor pane is a path bar showing the relative location of developer current file. Click any location in the path to immediately jump to any other file at that level. This is the Jump Bar, and it is so efficient that may wish to dedicate entire desktop to source code, quickly jumping from file to file.

2.6.4. Interface Builder is Built-in

Selecting an interface file (.nib/.xib) in project will open the IB editor within Xcode. Opening the right-hand Utility area will show the full complement of interface inspectors, as well as the library of controls and UI objects. Drag a control from the library, and drop onto the canvas, to layout Mac OS X or iOS application. The Interface Builder is Built-in are show in Figure 2.17

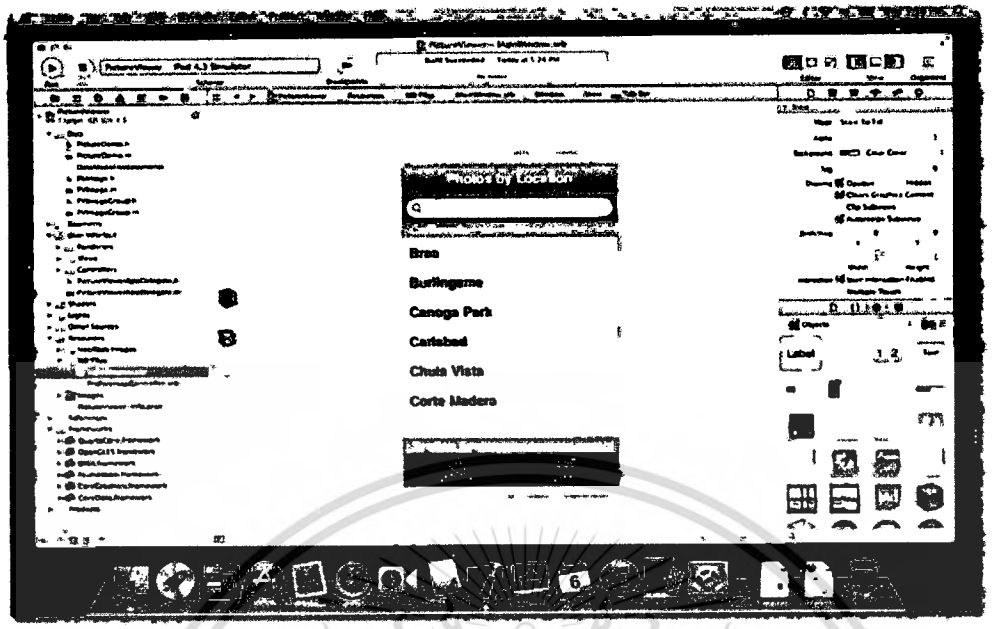


Figure 2.17: Interface Builder is Built-in.

The developer can drag connections directly from the UI design to the source code. Xcode 4's new split editor layout makes it easy to wire up your actions and outlets simply by dragging a connection to existing code.

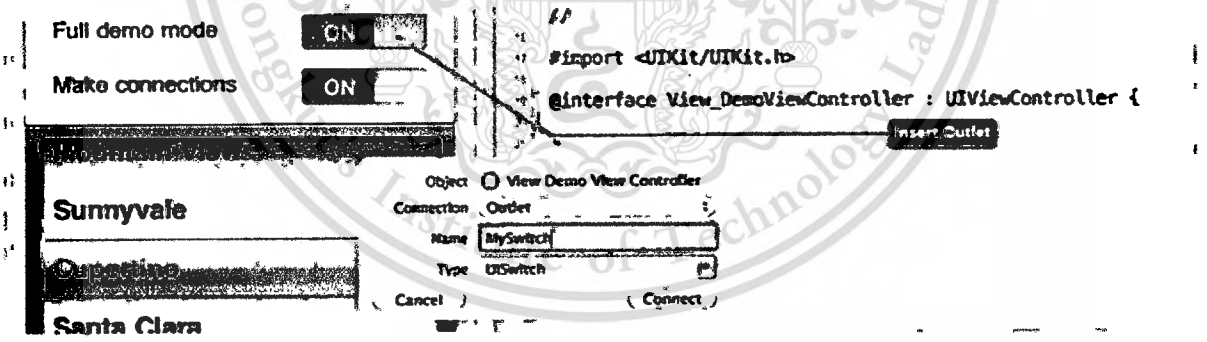


Figure 2.18: Drag and connection.

2.6.5. Assistant

With the new Xcode Assistant, the two-pane editor layout in Xcode 4 becomes dramatically more powerful. When turn on Assistant, the IDE will anticipate which other files that need to see, as developer work. Editing a new derived class? The Assistant will show the code for the class are inheriting. Writing new implementation code? The Assistant will automatically show the corresponding header. When designing an interface, the Assistant will show the appropriate controller, making drag-and-drop code connections extremely simple. Data model designing will bring up the classes that back the models — all automatically.

2.6.6. Apple LLVM Compiler 2.0

Apple LLVM is the next-generation compiler technology powering Xcode 4. Based on the vibrant open source LLVM.org project led by Apple engineers, the Apple LLVM compiler is modern thinking, tuned for iPhone, iPad, and the multi-core Mac.

Apple LLVM is fast. It compiles code twice as quickly as GCC, yet produces applications that also run faster. The compiler was built from the ground up as a set of highly optimized libraries, easy to extend, easy to optimize, and designed for today's modern chip architectures. In Xcode 4, the full Apple LLVM compiler stack — from the front end parser, to the back end code optimizer — has great support for C, Objective-C, and C++.

Syntax highlighting, code completion, and every other index-driven feature is handled by the LLVM parser. If the compiler knows about a symbol, so does the Xcode IDE. C, C++, and Objective-C are all accurately understood at editing time, exactly as they are when building.

2.6.7. Fix the problem automatically

The IDE is intelligent enough to fix the problem. In many cases Xcode will not only report an error, it will present a solution as well. Click the error to see the available Fit-its, such as correcting an assignment to a comparison, repairing a misspelled symbol, or appending a missing semicolon. A single keyboard shortcut will instantly have the error repaired, and let you continue coding.

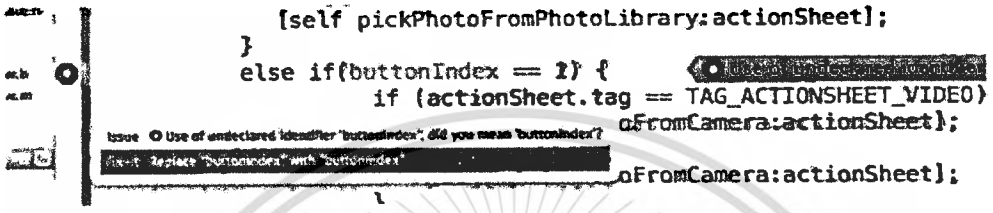


Figure 2.19: Fix the problem automatically.

2.6.8. Version Editor

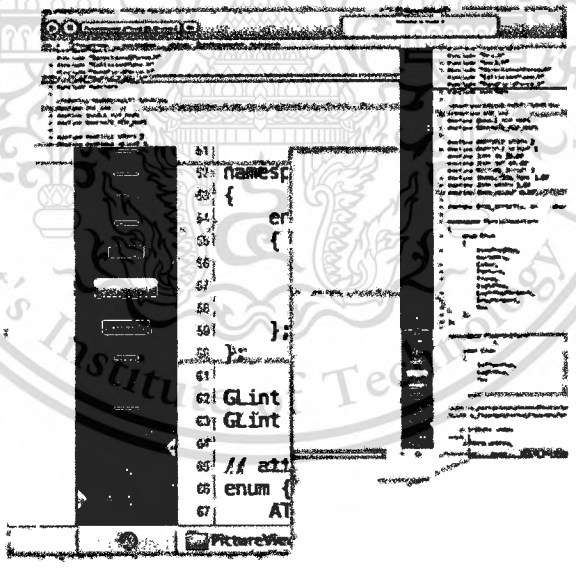


Figure 2.20: Vision editor.

The new Version editor in Xcode 4 makes it easy to see any two versions of source code, side by side, in a live editor. More importantly, the Version editor is a new way to think of source control management in an IDE, because the comparison view is also a timeline. Drag the slider in the middle and travel back in time through your project, comparing any two versions.

The Version editor can also show you a detailed log of past events, and track blame for past check-ins. Complex SCM commands are managed behind the scenes. It is even possible to manage multiple projects within a single Xcode 4 workspace, one project managed in Subversion, the other in Git, all updated automatically.

2.6.9. Debugger

Xcode 4 introduces LLDB, a brand new debugging engine contributed by Apple to the LLVM.org open source project. Like LLVM, the new LLDB engine is designed from the ground up to consume much less memory, and be a rocket when it comes to performance.

The new LLDB debugging engine is the perfect fit for the new Xcode 4 debugging interface. When the application is running, the navigator will show a stack trace can expand or compress to show or hide stack frames as you debug. As step through, can even lock onto a single thread then click "continue" and follow that specific thread of execution. Multicore debugging in Xcode 4 is now as easy as multicore coding with blocks and Grand Central Dispatch.

2.7 Introduction to Unity

Unity is an integrated authoring tool for creating 3D video games or other interactive content such as architectural visualizations or real-time 3D animations. Unity's development environment runs on Microsoft Windows and Mac OS X, and the games it produces can be run on Windows, Mac, Xbox 360, PlayStation 3, Wii, iPad, iPhone, as well as the Android platform. It can also produce browser games that use the Unity web player plugin, supported on Mac and Windows but not Linux. The web player is also used for deployment as Mac widgets. Unity also has the ability to export games to Adobe's Stage 3D functionality in Flash, but certain features that the web player supports are not useable due to limitations in Flash.

2.7.1 An Integrated Editor

Unity provides an editing environment where the users organize the project assets, create game objects, add scripts to these objects, and organize objects and assets into levels. Most importantly, Unity provides a "game" view for the content. Users can hit play and interact with the content while they watch values, change settings, and even recompile scripts.

The IDE is largely stateless, in that there is little distinction between creating the levels and playing. For example, the editor remains functionality identical whether the content is stopped or currently playing. This is hugely useful, because while the content is playing users can hit pause and then move things, create new objects, and add scripts

2.7.2 Component Architecture Paradigm

Unity utilizes a component-based architecture. Users could ignore this in creating game logic. In Unity, every object in the scene is a GameObject. An arbitrary number of Components are attached to GameObjects to define their behavior.

For example, a physical crate might be:

- GameObject (name, layer, tags)
  - Transform (position, rotation, scale, parent)
  - Mesh Renderer (actually draws the object)
  - Box Collider (define collision volume)
  - Rigidbody (movable physics object)

2.7.3 Game Engine

Unity is a fully-featured game engine. It includes and exposes many systems needed for game creation, such as:

- Graphics Engine

  Unity's graphic engine includes a shader language, ShaderLab, which wraps Cg and GLSL shaders with additional engine semantics.

- Physics Engine

  Unity uses NVIDIA PhysX for their physics engine, with editor and API integration (set up collision volumes, joints, and things by adding components to your GameObjectsin the editor, and script physics with things

like Rigidbody.AddForce() and MonoBehaviour. On Collision Enter() callbacks).

- Audio Engine

  Unity has a positional audio system. The users can play sounds in 3D space, or "2D" stereo sounds.

- Animation System

  Unity includes an animation system, including support for animation layers, blending, additive animations and mixing, and real-time vertex/bone reassignment.

  There are quite a few other out-of-the-box systems to help, too, like particle systems, networking, UnityGUI, and so on.

2.7.4 Scripting Platform

Unity embeds Mono to power its scripting environment. The developer can script in C#, JavaScript, or Boo (a Python variant). Mono itself is an open-source version of the .NET development environment. Note that this does not mean that Unity requires .NET. Mono is totally distinct from Microsoft's .NET, and Unity totally embeds Mono.

It's also worth noting that Unity's use of Mono goes above and beyond the compiler and common language runtime. The developer also get the full .NET namespace, which means a huge amount of classes are available to out of the box: XML parsing, cryptography, sockets, and more.

2.8 UVLayout Pro

UVLayout is a stand-alone application for the creation and editing of UV coordinates for 3D polymeshes and subdivision surfaces. Used by professionals in the games and visual effects industries, by hobbyists of all ilks and by students, UVLayout's unique approach gives texture artists the tools to produce high quality low distortion UVs in significantly less time than they would by traditional methods.

Using UVLayout could be described as the opposite of dress making; instead of cutting out a flat pattern and sewing that up to make clothing, in UVLayout the object is cut into pieces that are then flattened out to make the pattern. These flattened UV shells are not just planar projections; a dynamics based algorithm is used to spread the UVs out, as you watch, so that there's minimal stretching, compression or skewing of textures when they're applied to the object.

Some major features of UVLayout are:

- **OBJ** import and export
- **Edge-loop Detection** for quicker UV seam selection
- **Symmetry Editing** for faster flattening of symmetrical meshes
- **Edge Straightening** on shell boundaries and interiors
- **Flattening Brushes** for local tweaks of the automatically generated UVs
- **Auto Packing** of UV shells to minimize wasted texture space
- **Auto Stacking** of similar shells for shared texture space usage
- **Subdivision Surface** calculations based on limit surface shape
- **Unlimited Undo** of all editing functions
- **Plugin Interface** for integration into other applications

Chapter 3

Design and Implementation

3.1 Principle

From the theory and principle of AR processing, which can extract the marker and calculate the position and coordinate of marker for represent to the singers 3D models on the marker. In this chapter will cover overview of the system, the singer 3D models, and the role of String library.

3.2 System overview

3.2.1 Functional

3.2.1.1 Choose the singer 3D model.

The user can choose the singer that they interested from the menu.

3.2.1.2 Choose the song.

After user chooses the singer that they interested, they can choose the song to listen from the list, that including song name belong to the singer.

3.2.1.3 Choose the cloth.

User can choose the singer cloth before listen to the music. The application provide the example of cloth image, user can choose the cloth by touch on it. The singer model can ware T-shirt, Jacket, and etc.

3.2.2 Non functional

3.2.2.1 User Interface easy to use. User can understand how to use the application by themselves.

3.2.2.2 User enjoyed the AR application in real time.

3.2.2.3 User has alternative choice to choose the entertainment technology for listen to the music. They can enjoy visual and audio simultaneously.

3.2.2.4 In term of business, the Organization can use this application to apply with their business, and make more benefit to them.

3.3 Step for implementation

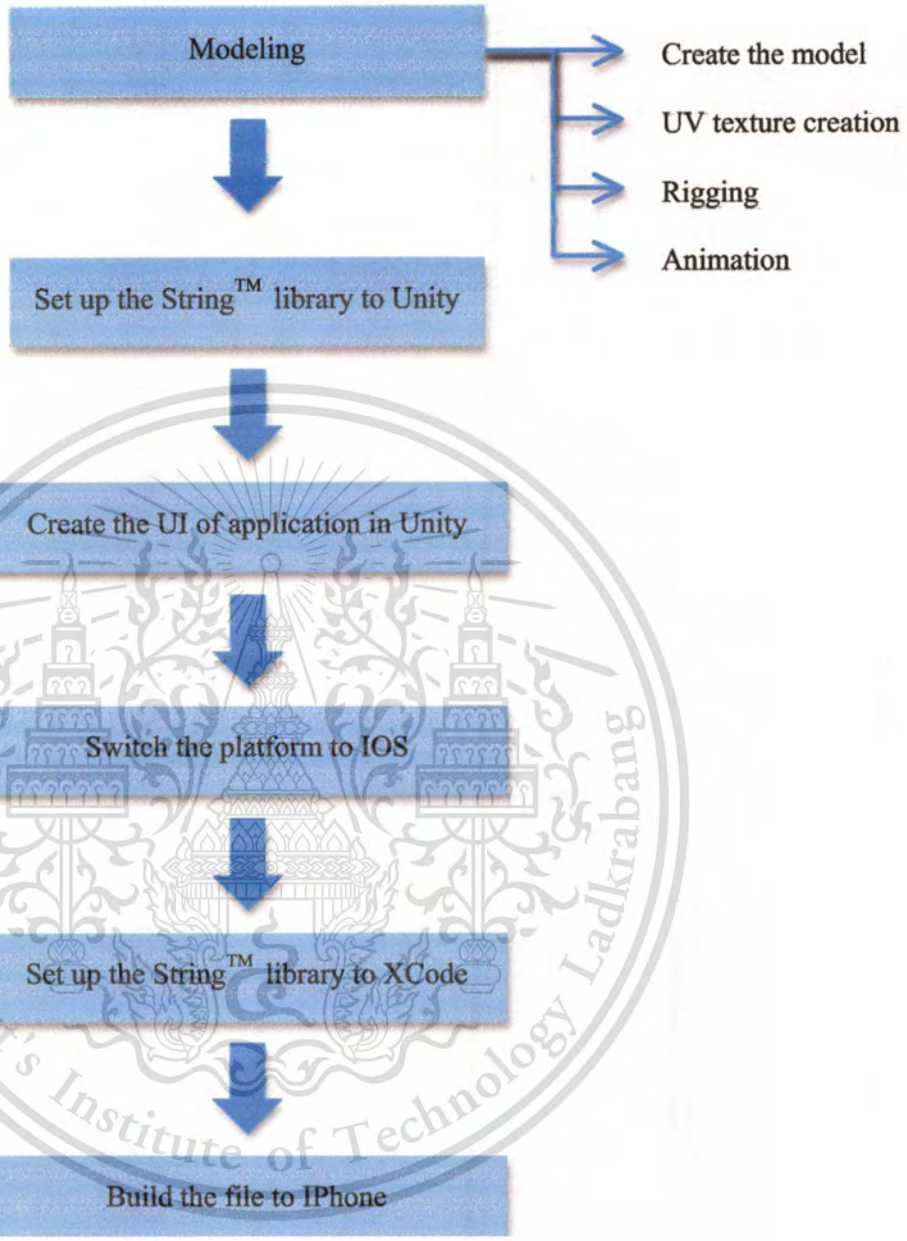


Figure 3.1: Step for implementation.

3.4 The singer models creation.

Creating model developer by using Autodesk Maya 2012, this method is used to create the 3D singer models and animations. By using UV Layout Pro creates UV Texture, and Adobe Photoshop paints UV Texture of the singer's cloths.

3.4.1 The 3D singer model creation.

3.4.1.1 Open Autodesk Maya 2012.

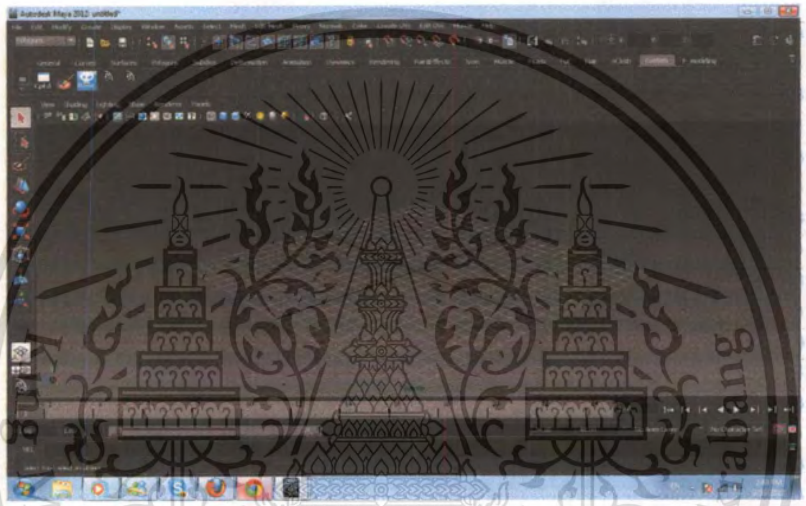


Figure 3.2: First screen of Autodesk Maya 2012.

3.4.1.2 Set up the project folder. Sort the data by type, then click on the file, and choose the project set.

Figure 3.3: Set the Project for sorting the data

3.4.1.3 Select the directory to place the project folder, and then click to set the button. The project folder will be created into the location that the user sets. The project folder will be separated into each data type, and it is easier for the user to stall the data.

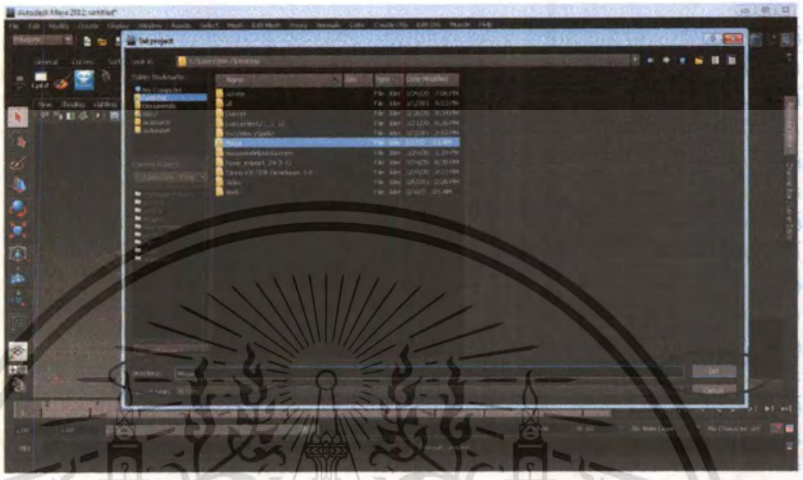


Figure 3.4: Set Project screen of Autodesk Maya 2012.

Figure 3.5: Folder Project of Autodesk Maya 2012.

3.4.1.4 When finished creating, the folder project will go back to the Autodesk Maya 2012. Click the Image Plane to replace Bird picture on the plane. Choose the picture of Bird, and then click Open.

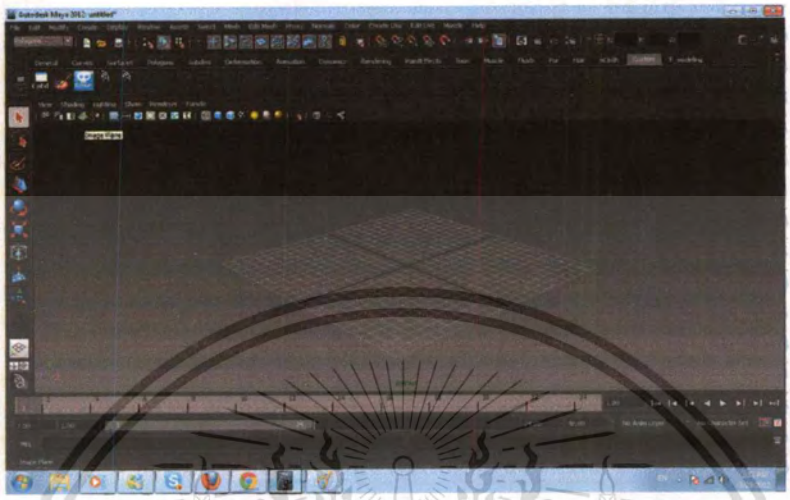


Figure 3.6: Click to Image Plane for replace the picture.

Figure 3.7: Select picture.

3.4.1.5 After added a picture to the image plane, then create the polygon for the head of the model.

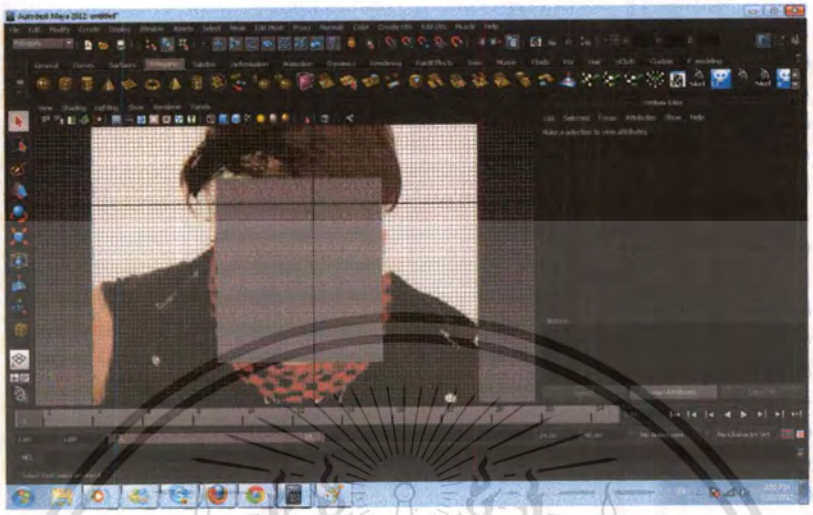


Figure 3.8: Create polygon forward singer photo.

3.4.1.6 In this project, we create in half the part of the polygon, and reduce the vertex to match with Bird face. Cut the polygon into half by using insert edge loop tool from Edit Mesh function. Select to the center of polygon, and then separate into half.

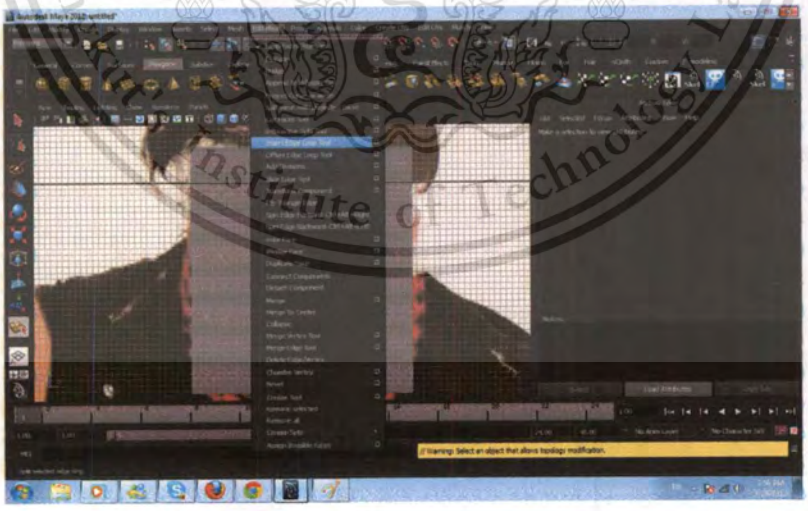


Figure 3.9: Use function Insert Edge Loop Tool to separate polygon into half.

3.4.1.7 After use Insert Edge Loop Tool, then right click to select the Face for cutting half polygon, and press delete button.

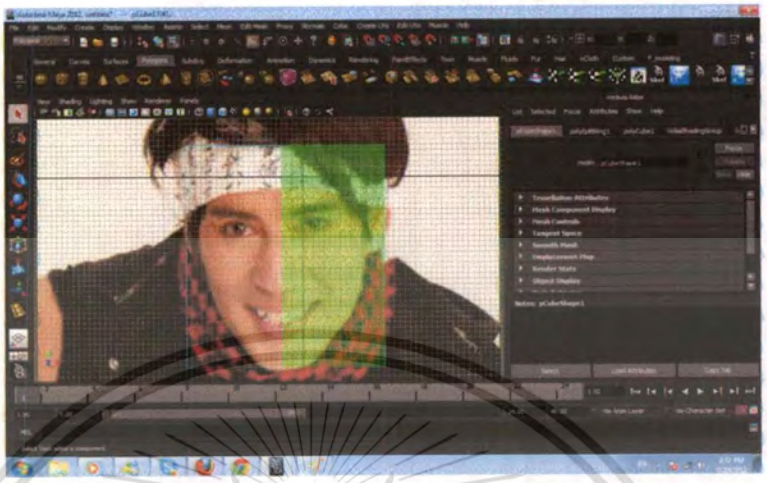


Figure 3. 10: Select half Face of polygon and delete.

3.4.1.8 When finished deleting half face of the polygon. Click right and choose vertex for manipulate the vertex point by pressing W for translate. Press E for the scale, and press R for Rotate. Then adjust the vertex of polygon like Bird face.

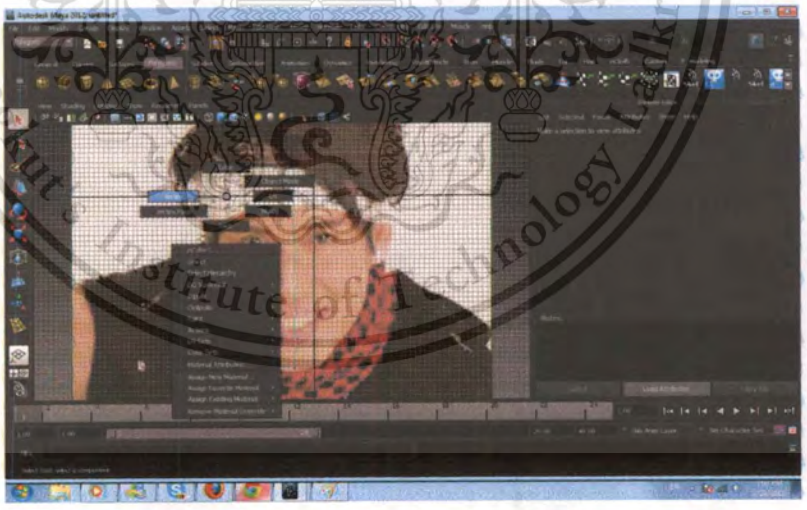


Figure 3.11: Click right and choose vertex for manipulate.

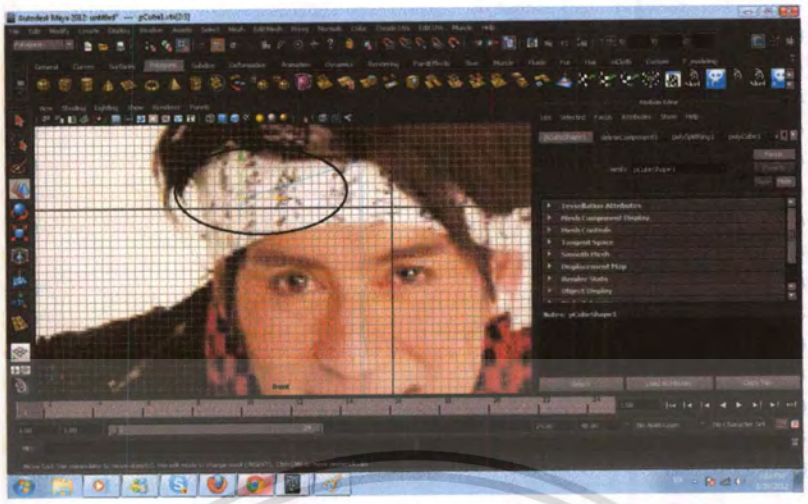


Figure 3.12: Adjust vertex by Translate, Rotate, Scale.

3.4.1.9 Create the eye by using the Sphere. Create the mouth and the eyebrow by using the polygon.

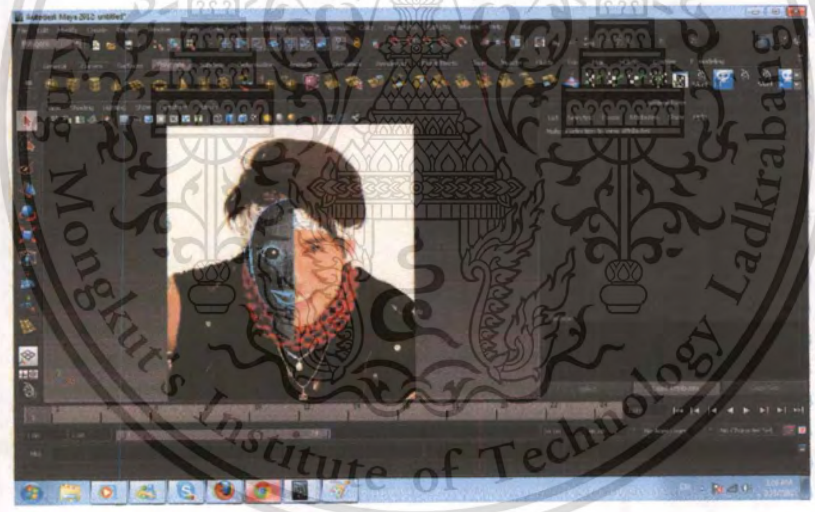


Figure 3.13: When duplicated all polygon of model finish.

3.4.1.10 After finished creating the head of the model, click at Mesh on the menu tab. Click at Mirror Geometry function, and choose Mirror Direction then click Apply.

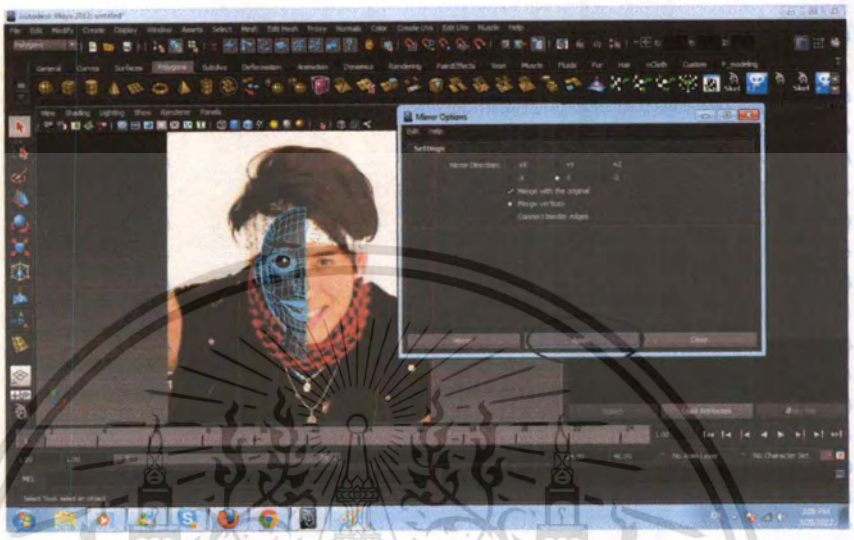


Figure 3.14: Mirror Geometry function.

Figure 3.15: Complete model from Mirror Geometry function.

3.4.1.11 When the head of model are completed. Create hair by using the polygon, and adjust the vertex of the polygon.

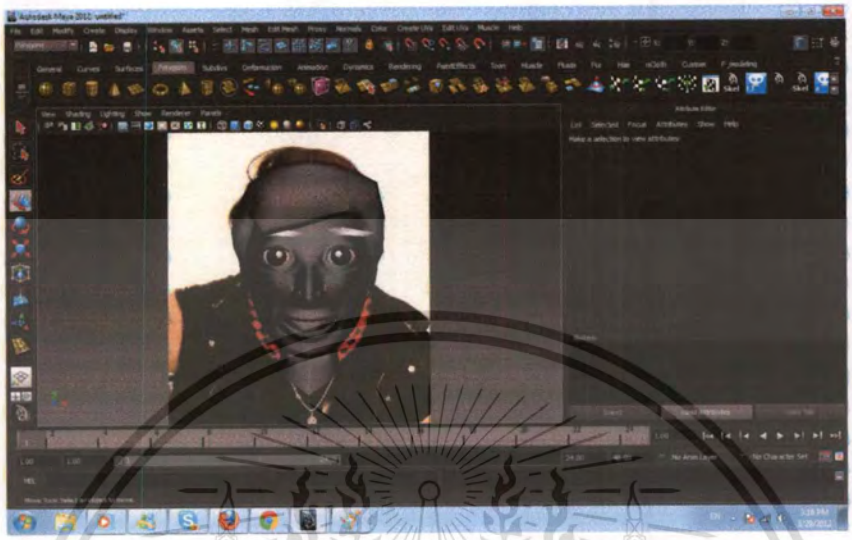


Figure 3.14: Complete hair by use polygon and adjust vertex.

3.4.1.12 After finished creating head. Next create the shirt, pant, body, arm, leg, shoe and their accessories in one side by using Mirror Geometry function again.

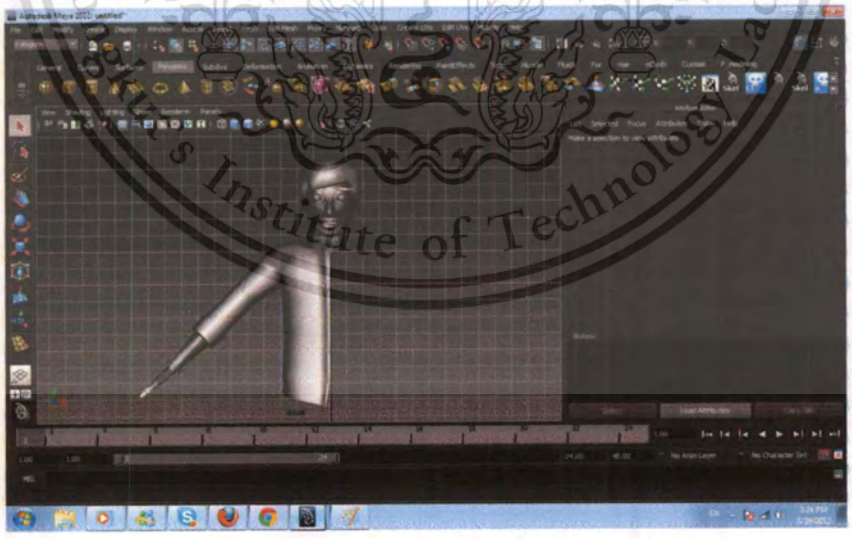


Figure 3.15: Create singer's shirt and arm in one side.

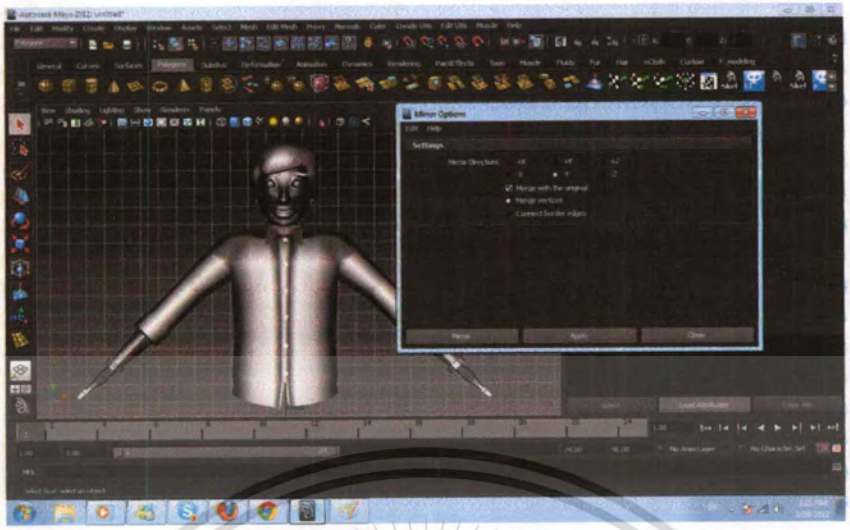


Figure 3.18: Mirror Geometry function.

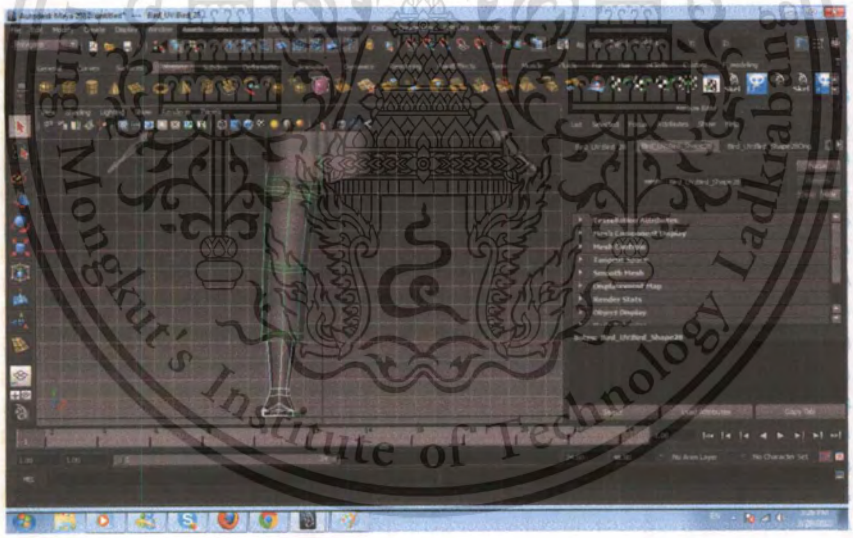


Figure 3.19: Create singer's pant and shoe by polygon in one side.

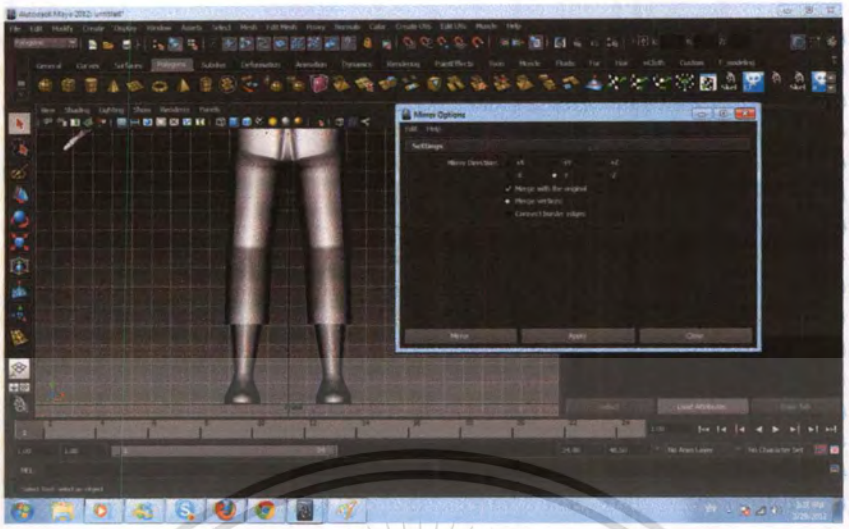


Figure 3.20: Use Mirror Geometry function for their pant and shoes.

3.4.1.13 When finished creating singer model, the next step is to assign color material to the model. Click right by choosing object mode, and then select which polygon that needed to assign color. Click right on mouse and choose Assign favorite Material. Click to lambert.

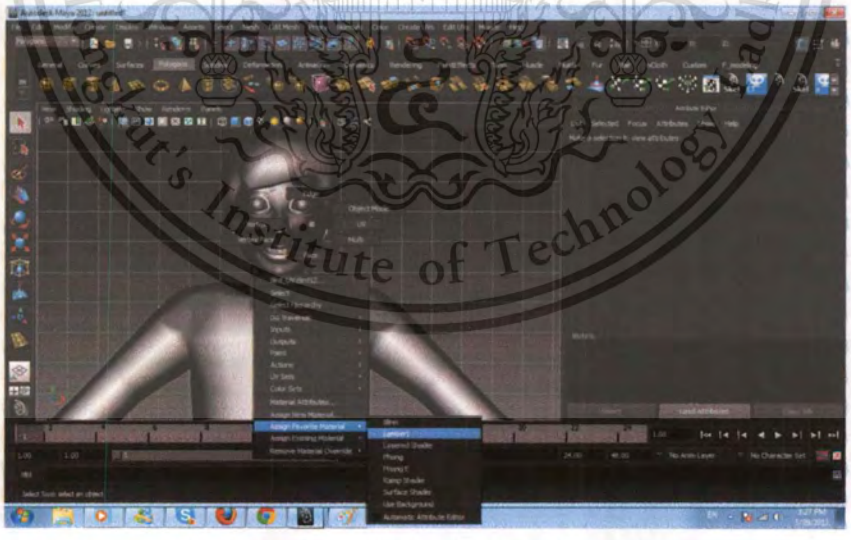


Figure 3.21: Assign color material option.

3.4.1.14 After click to lambert, the option of this material will show in the right tab of program screen. Click at the material color to select color, which assign to the model.

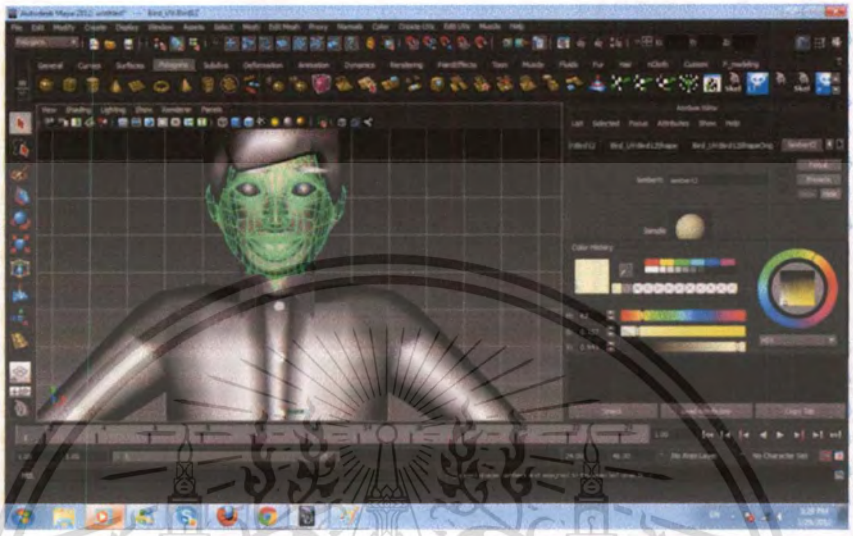


Figure 3.22: Assign material color to head of model.

3.4.1.15 Repeat each step in order to assign the color to the polygon on each model.

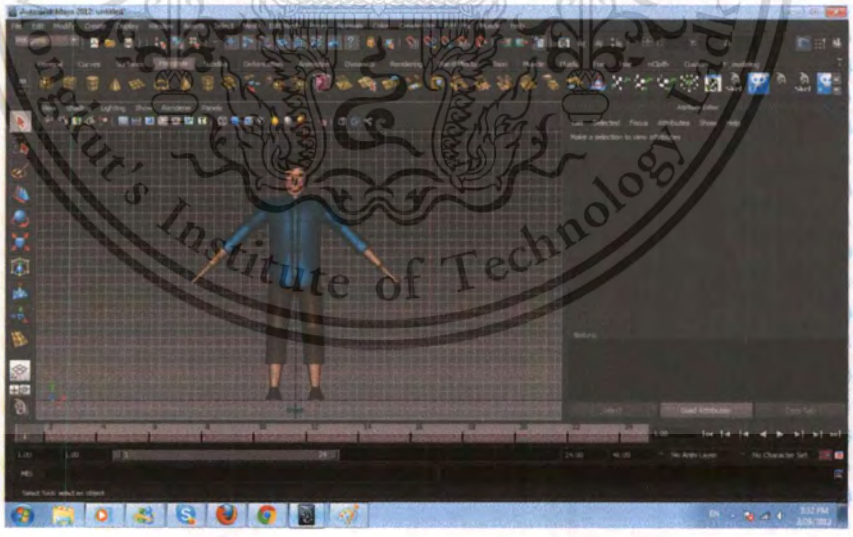


Figure 3.23: Model is complete to assign color material.

3.4.2 Create UVtexture.

3.4.2.1 Select the objects that you wanted to create the UVtexture then export them to .OBJ file.

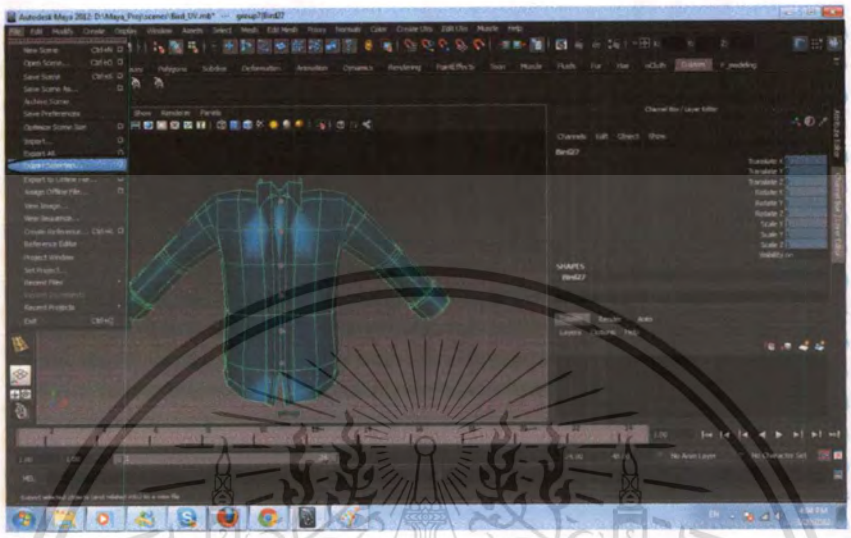


Figure 3.24: Select the object and export to .OBJ file.

3.4.2.2 Open the folder that have an object in .OBJ file. Drag and drop the file to UVlayout program.

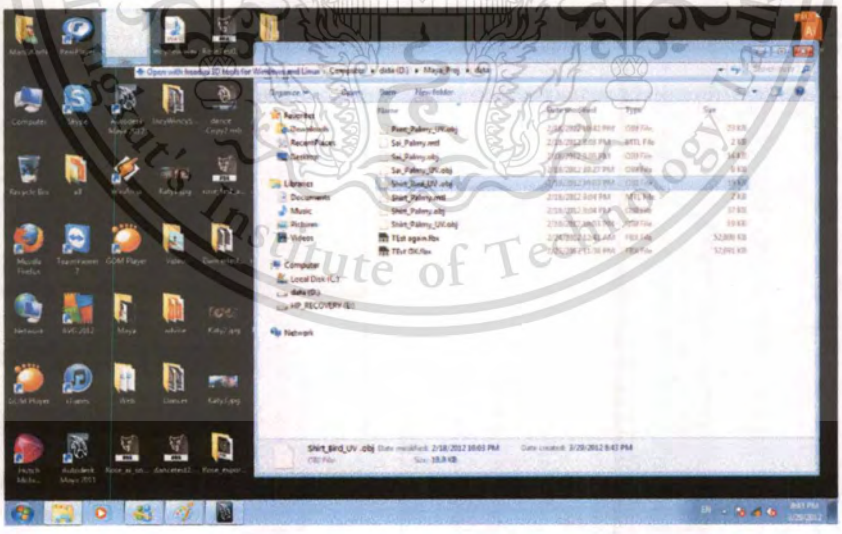


Figure 3.25: Drag and drop file that want to create UVtexture to UVLayout.

3.4.2.3 When drop the file to UVlayuot program. The program will open, and the object that wanted to create UVtexture. It will show on the program screen.

Figure 3.26: Object that want to create UVtexture show on program screen.

3.4.2.4 Click on Edit menu to show options and click find. The program will find the detail of the object that prepares to create UVtexture.

Figure 3.27: Edit menu of program UVLayout.

3.4.2.5 Click on the line and press c to select. Press enter to separate. When select the line, it will show in red color.

Figure 3.28: When click on the line and press c it will show red color on the line that select.

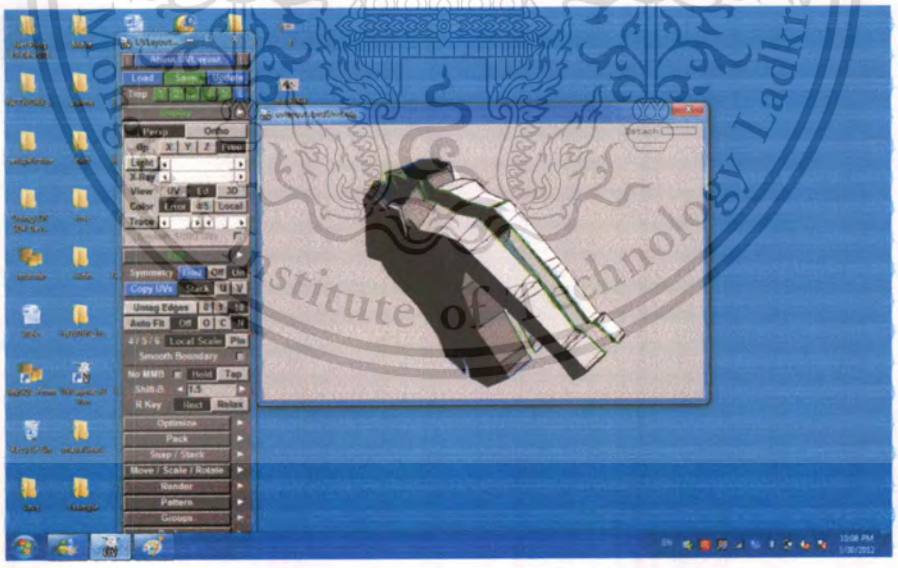


Figure 3.29: When press enter the object will separate.

3.4.2.6 After separate the object, and then click on 3D button in Display menu.

Figure 3.30: Click on 3D button for spread out the object to UVtexture.

3.4.2.7 Click UV on the Display menu to separate the object.

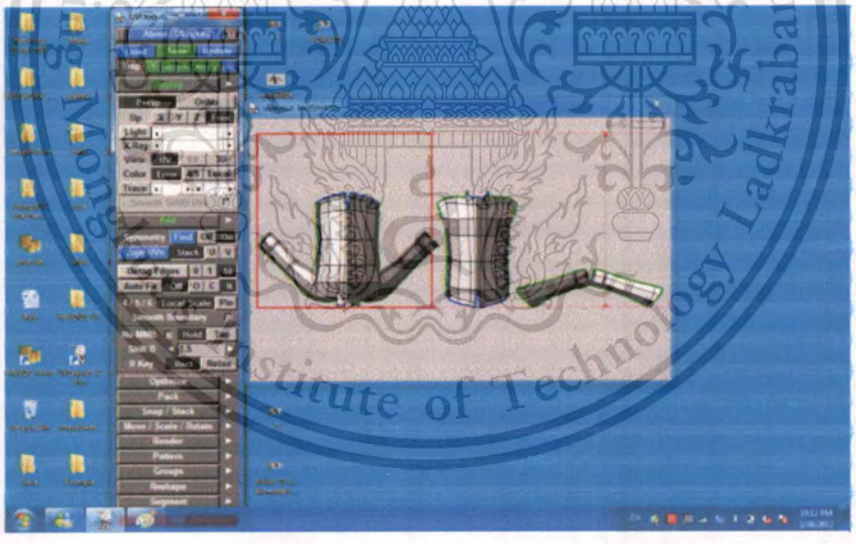


Figure 3.31: Click on UV button to separate object.

3.4.2.8 Select object press shift and F, this to spread out the object into UVtexture. Wait until it finishes.

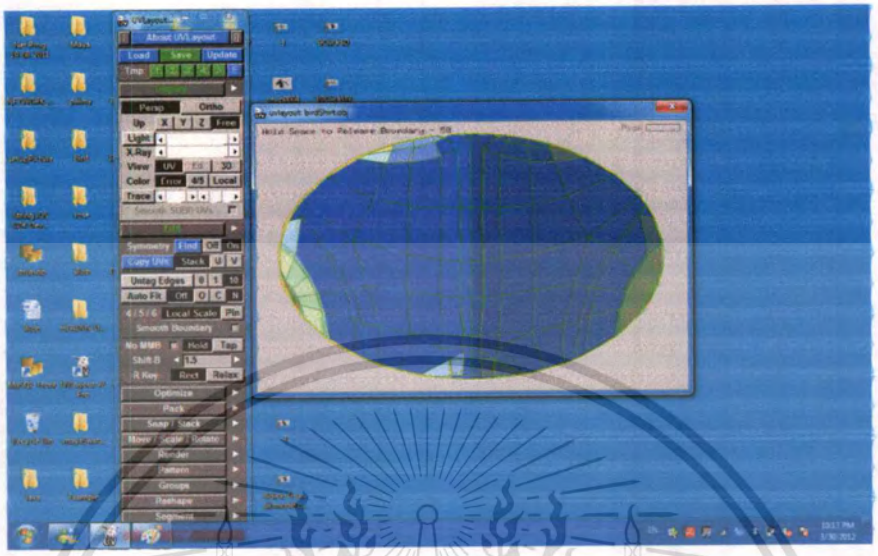


Figure 3.32: Spread out the object into UVtexture.

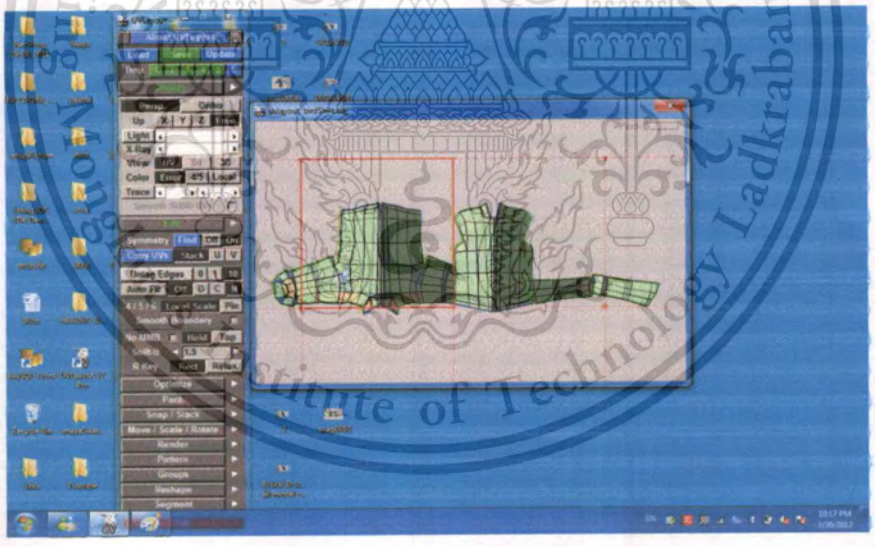


Figure 3.33: Finish from spread out the object.

3.4.2.9 When finish from spreading out the UVtexture. Click on SAVE menu, save to .png file.

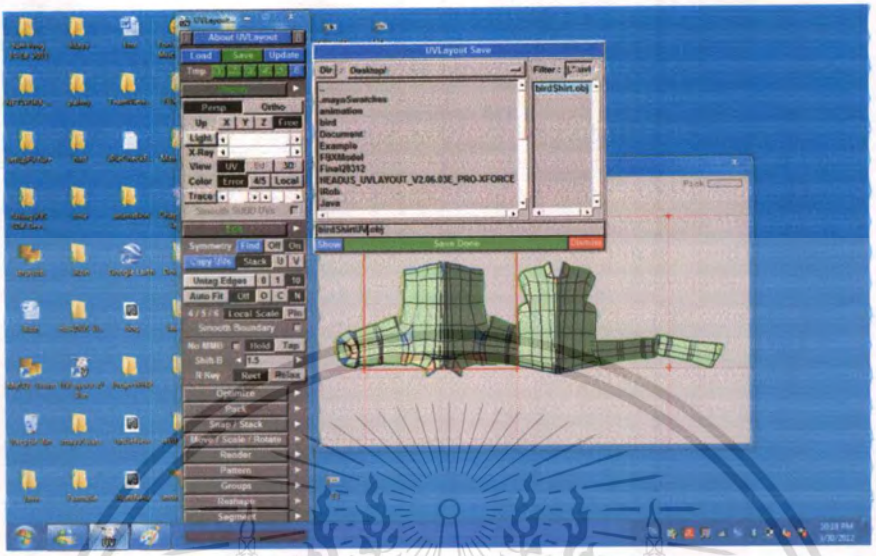


Figure 3.34: Save to .png file.

3.4.2.10 Open the old object, and import file. The .OBJ that used to spread out the UVtexture, replace it by the new object. Go back to the Autodesk Maya 2012. Open the singer model. Then delete

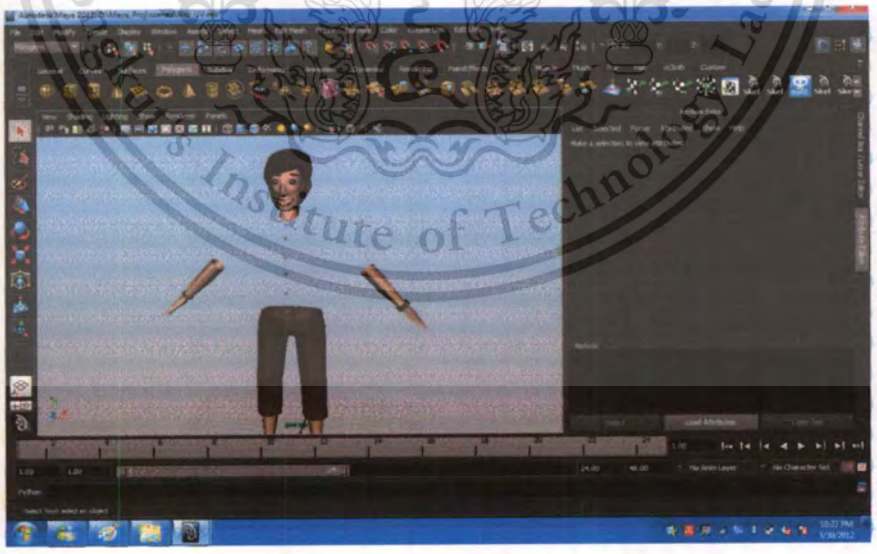


Figure 3.35: Open the singer model and delete old object.

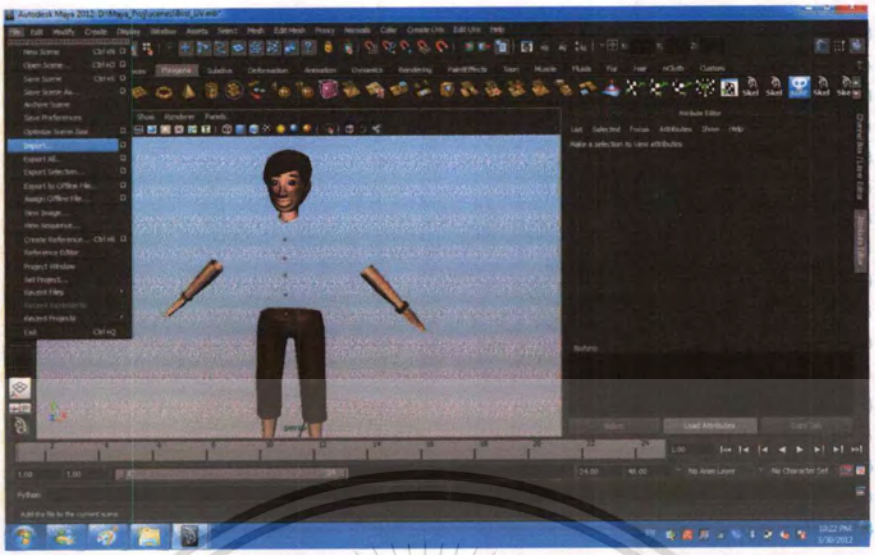


Figure 3.36: Import the new object that spread out UVtexture.

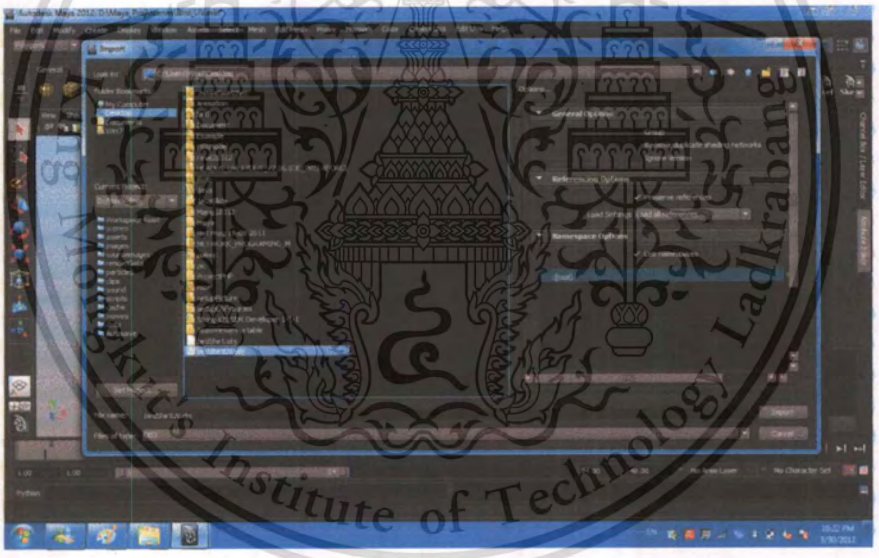


Figure 3.37: Select .OBJ file for import.

3.4.2.11 Click at the Window on the menu tab. Select at UV Texture Editor to create UV texture pattern.

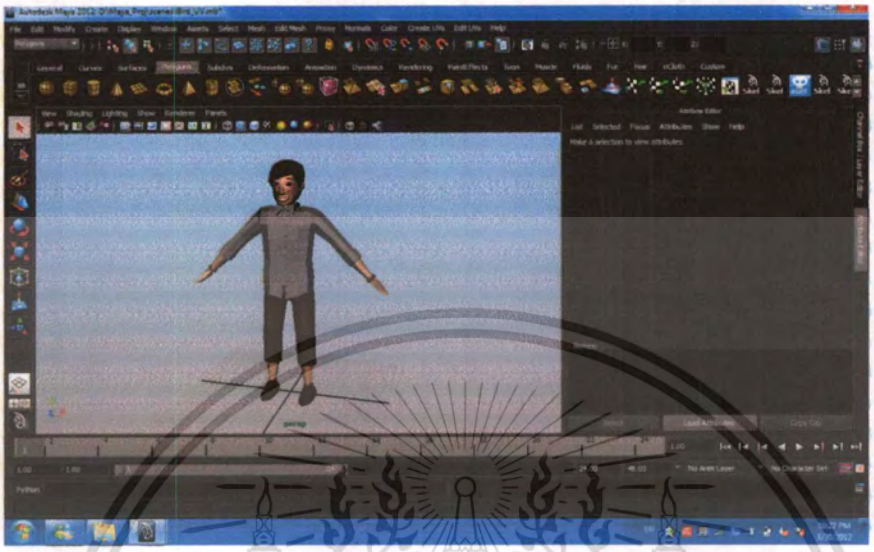


Figure 3.38: Replace new one to old position.

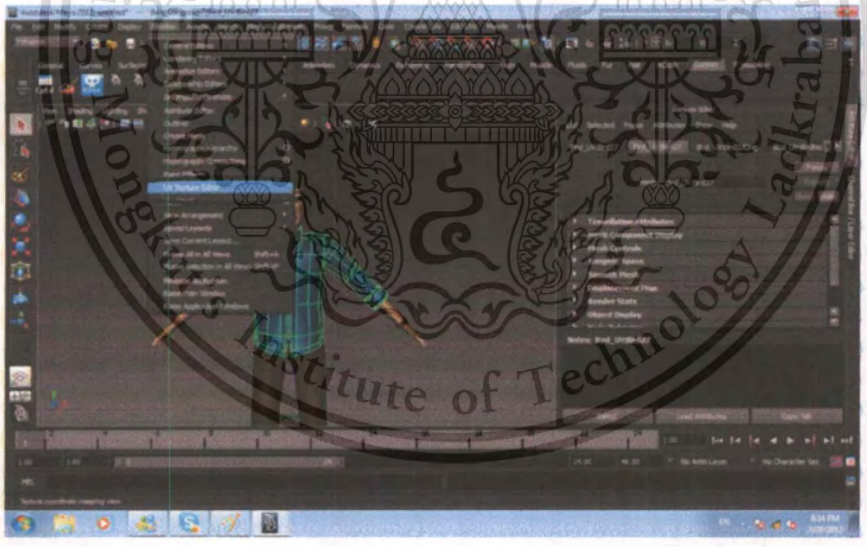


Figure 3.39: Choose UV Texture Editor for create UV texture pattern.

3.4.2.12 On UV Texture Editor. All the options will show on the screen. Click to select the object. UV texture will appear. You can arrange the position of shirt and pant into UV texture pattern.

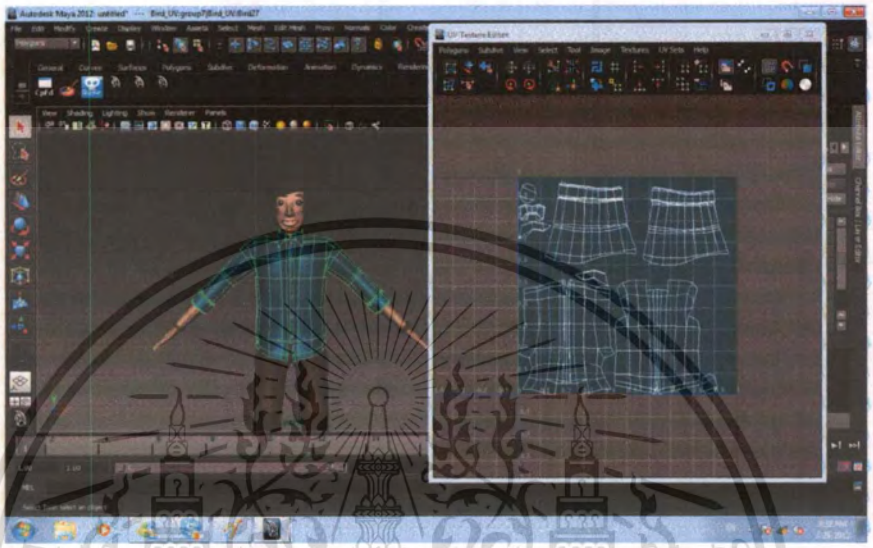


Figure 3.40: UV Texture Editor option.

3.4.2.13 When finish arranging, click on Polygon menu, and choose UV snapshot. This is to save UV texture pattern, the program will show UV snapshot option brows. This is the location to save and arrange size of X and Y for size of UV texture pattern. After that click OK to save.

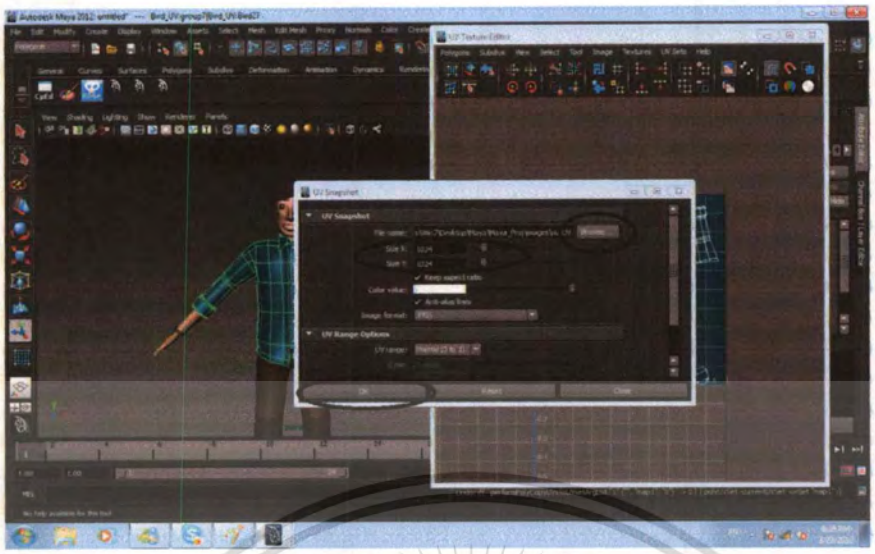


Figure 3.41: UV Snapshot option.

3.4.2.14 Open Adobe Photoshop CS5, open file from step 12 for paint UV texture. When finish, save the file to folder DATA in Maya project folder.

Figure 3.42: Paint UV texture in Adobe Photoshop CS5.

3.4.2.15 When painted UV texture finish, back to Autodesk Maya 2012 by matching UV texture to object. Click on the Window menu, choose Hyper shade. Click on Graph Materials at the Select Object. Follow from Figure 3.42, the material will show in Work Area.

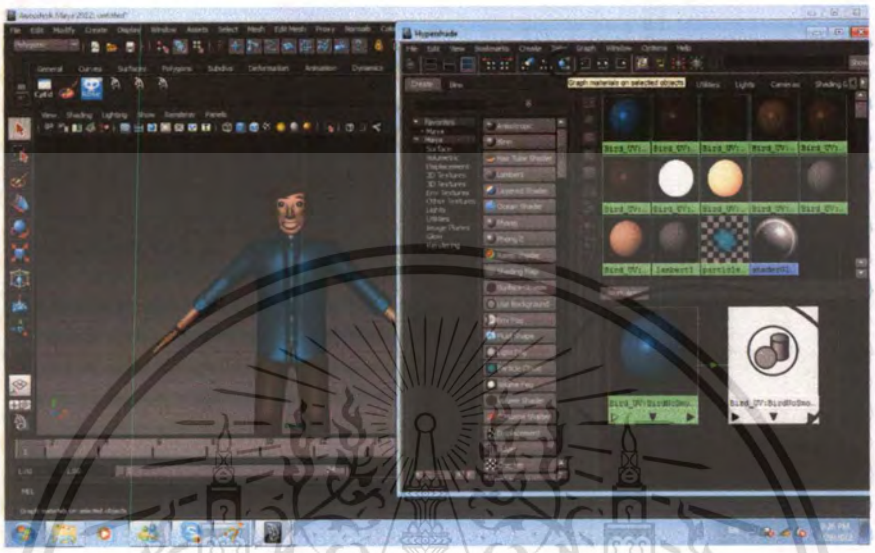


Figure 3.43: Hypershade option.

3.4.2.16 Click on color material in Work Area. Click option File from left side to assign material. Click no picture folder at the Image Name from option color material at the right side. Select UV texture in DATA file from Maya project folder.

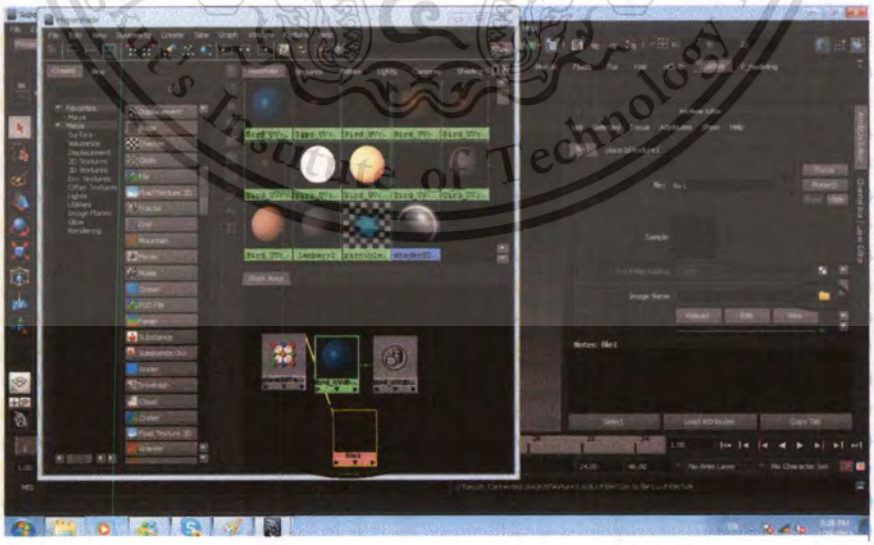


Figure 3.44: Select UV texture to assign.

3.4.2.17 When assign UV texture on color material, it will show the file in Work Area.

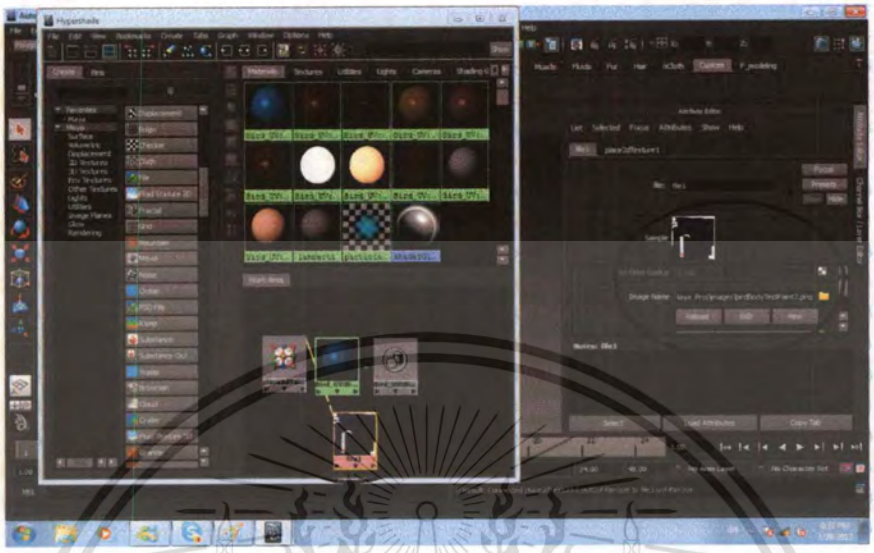


Figure 3.45: UV texture will show in Work Area.

3.4.2.18 Close all the options, press 6 button to see model assign at the UV texture.

3.4.3 Create Animation.

3.4.3.1 Before create animation, the model must have skeleton. In this special project the developer uses abAutoRig function to rig the model. Type abAutoRig; into command line in Autodesk Maya 2012(Figure 3.47) and press enter. After that, abAutoRig function will show on screen. Type the model's name in Char Name (Figure 3.48) and click Create Skeleton

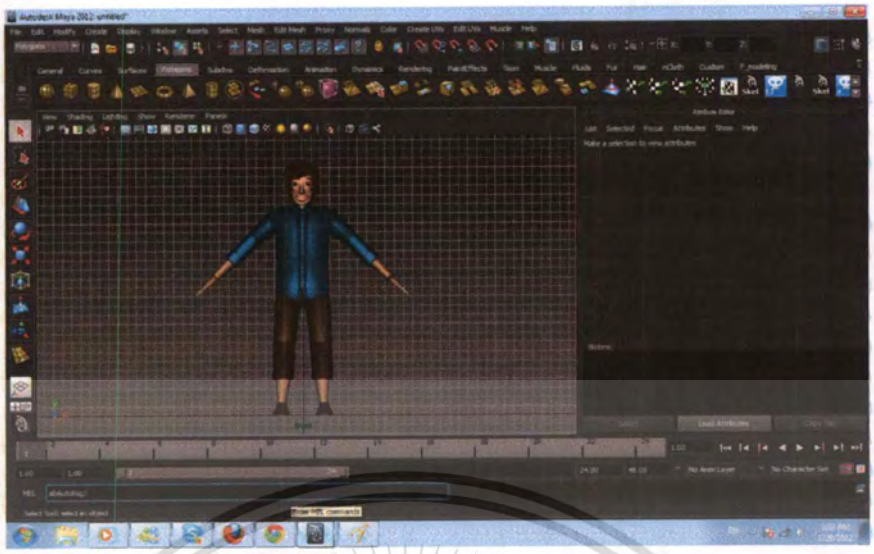


Figure 3.46: Use abAutoRig; function to rig model.

Figure 3.47: Type abAutoRig; into command line and press enter.

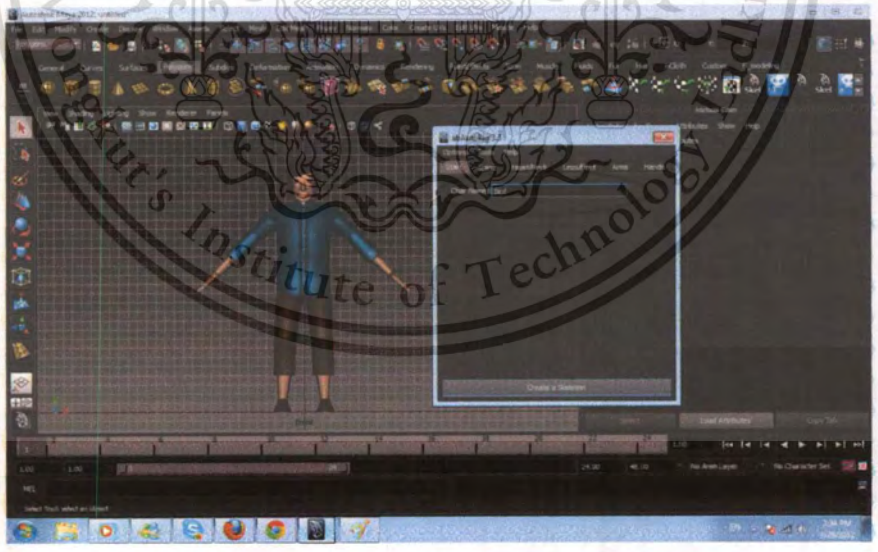


Figure 3.48: Function of abAutoRig will show on screen, type name on model and click Create Skeleton.

3.4.3.2 When create skeleton, the abSkeletonMaker option will show on the screen. Set the number of the finger on the model, and click Build Skeleton.

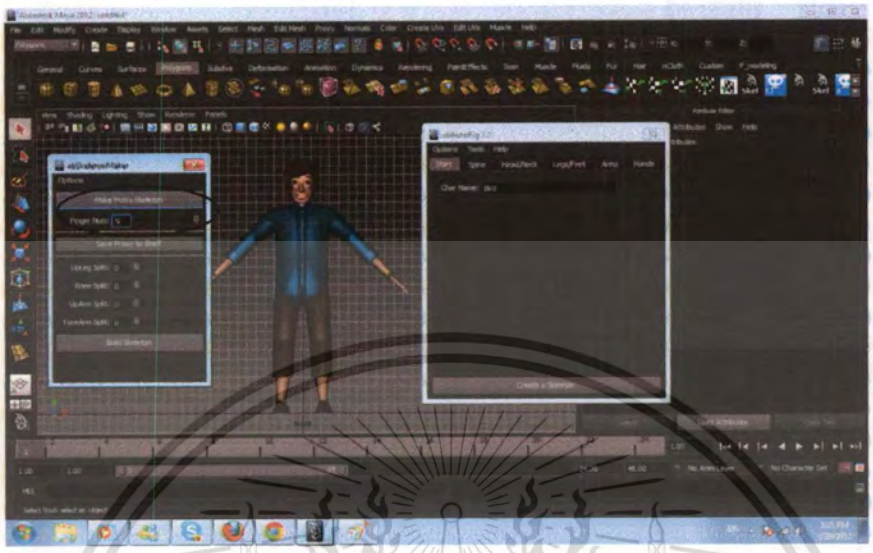


Figure 3.49: abSkeletonMaker option.

3.4.3.3 When click build skeleton, the skeleton will adjust into model, scale and rotate. The skeleton is fit to model by press E, R button. Using mouse to scale the skeleton, and click Build Skeleton.

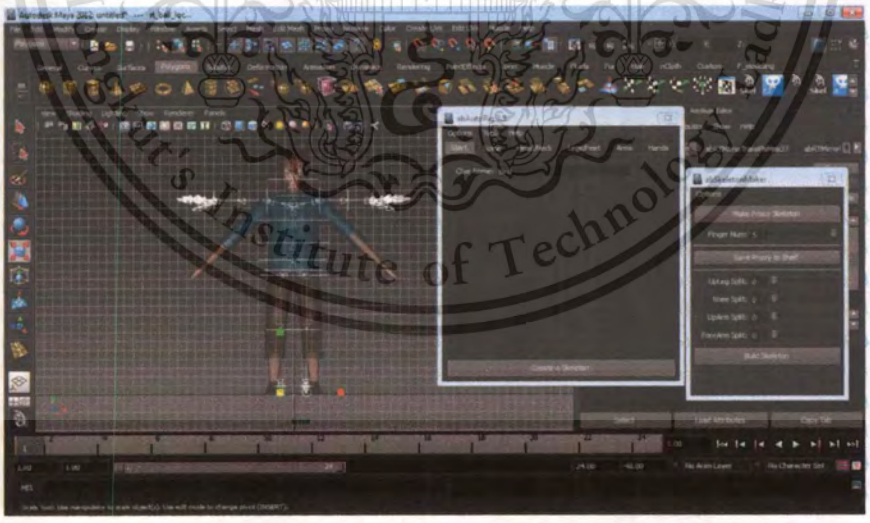


Figure 3.50: Scale, Rotate skeleton fit to model and click Build Skeleton.

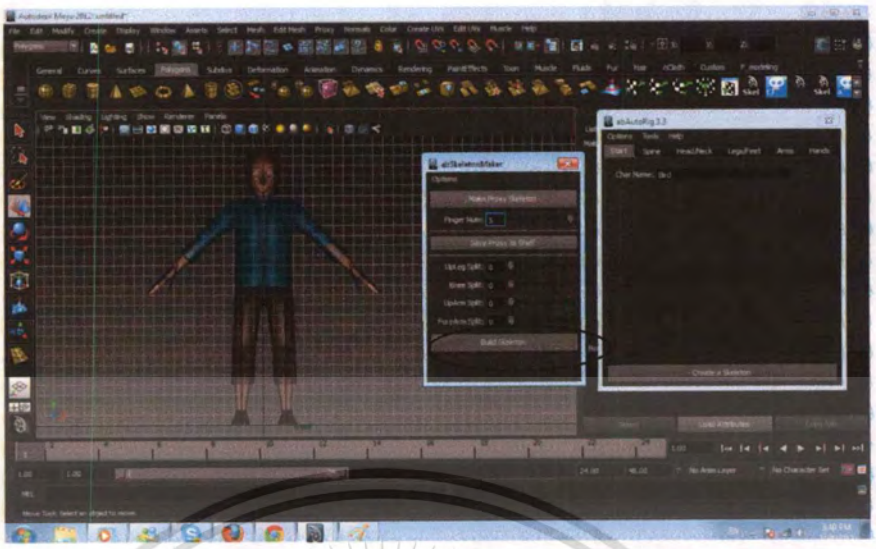


Figure 3.51: When click to Build Skeleton.

3.4.3.4 Click Spine tab and change the Spine Joint to 8, then click Create Spine Rig.

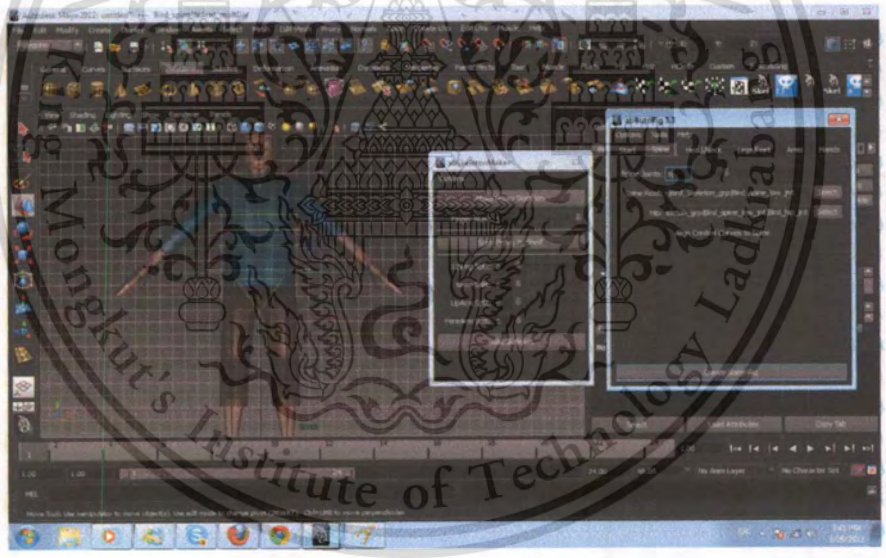


Figure 3.52: Create Spine Rig.

3.4.3.5 Click Head/Neck tab and then click Create Head Rig.

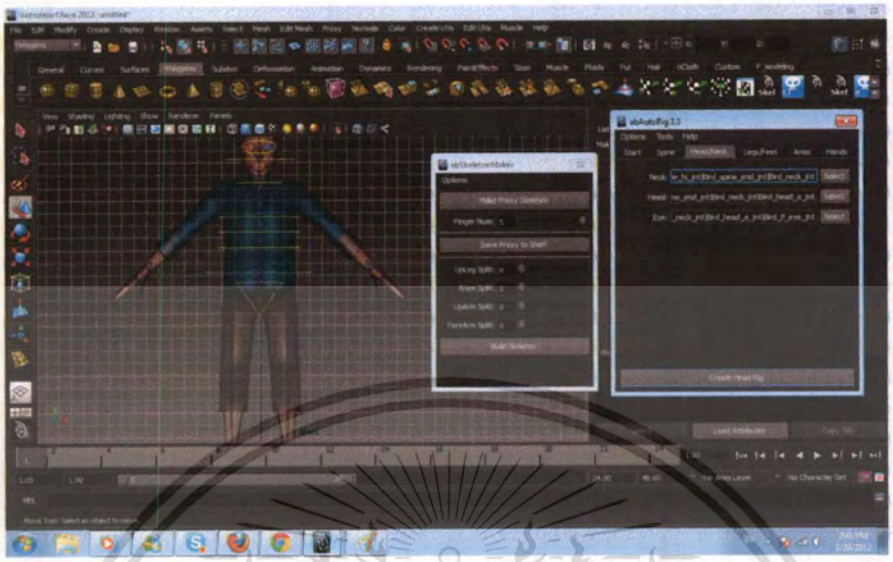


Figure 3.53: Create Head Rig.

3.4.3.6 Click Legs/Feet tab and click to Create Heel Locator, then click Create Leg Rig.

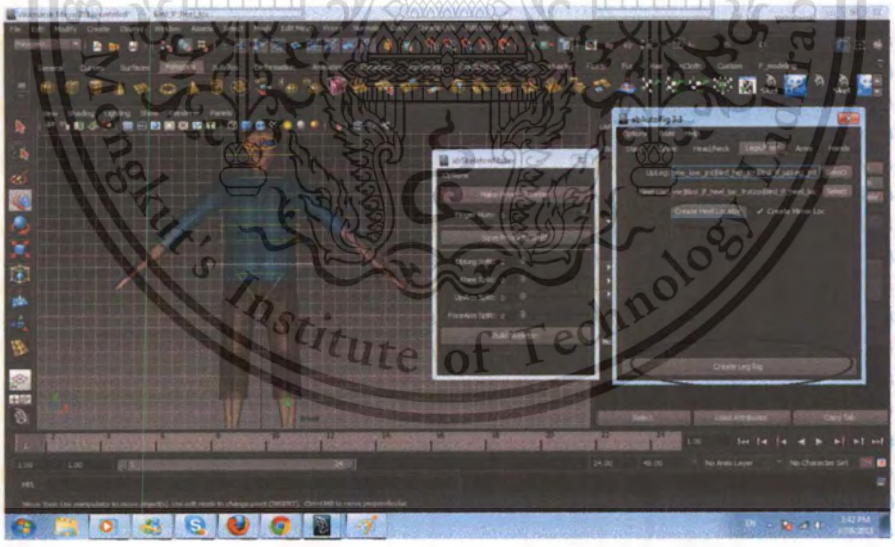


Figure 3.54: Create Leg Rig.

3.4.3.7 Click Arms tab and then click Create Arm Rig.

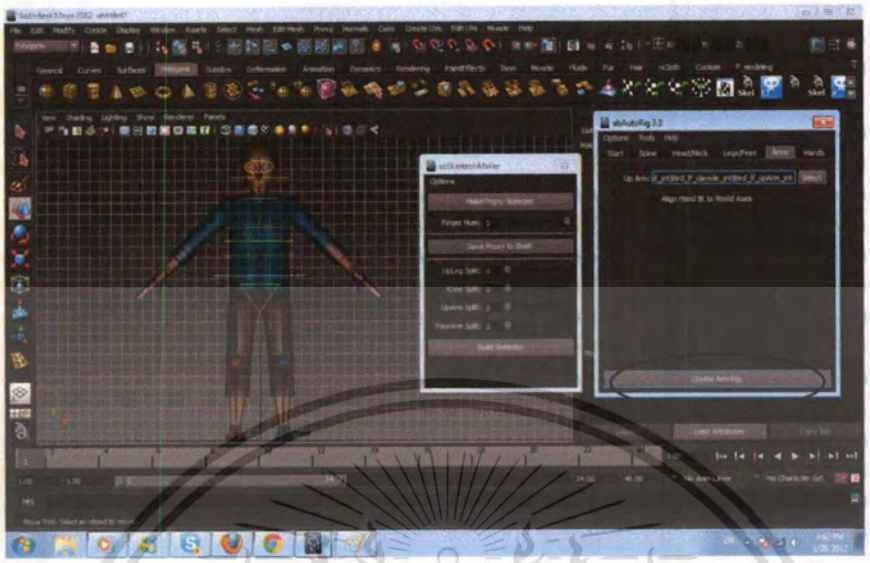


Figure 3.55: Create Arm Rig.

3.4.3.8 Click Hands tab and click Create Hand Rig. After that close all windows of abAutoRig function.

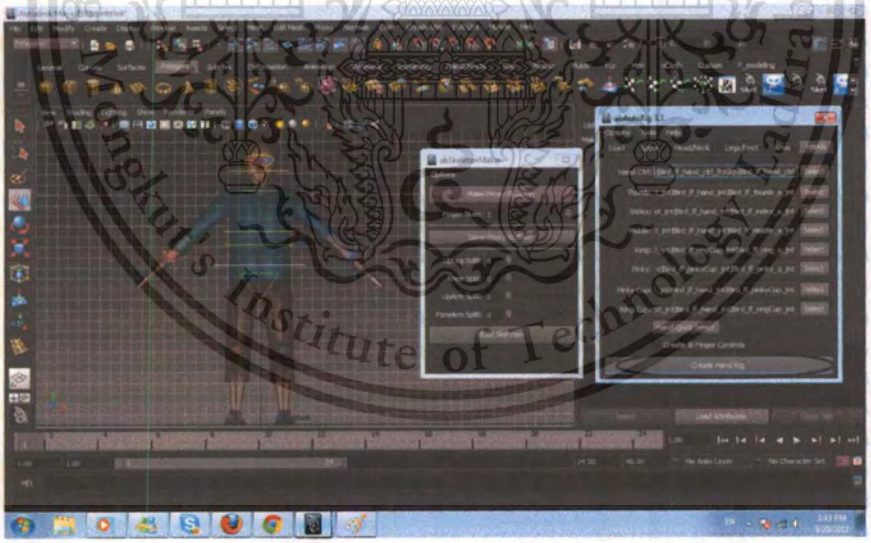


Figure 3.56: Create Hand Rig.

3.4.3.9 After finish from rig the model the rig control assign to model (Figure 3.58).

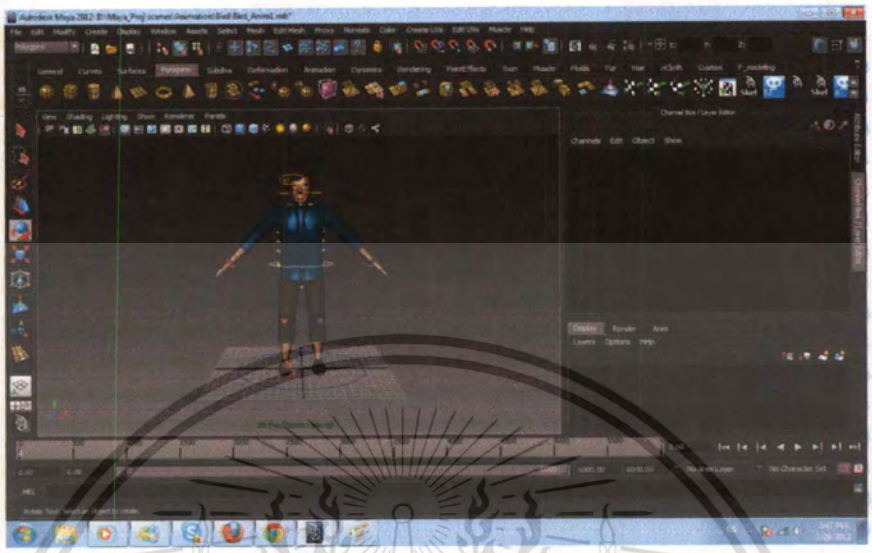


Figure 3.57: Rig Control assign to the model.

3.4.3.10 Click Script Editor for create MEL script, this script can help to create animation in easy way.

Figure 3.58: Create MEL Script from Script Editor.

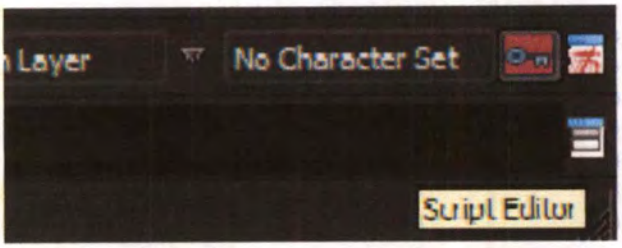


Figure 3.59: Script Editor function.

3.4.3.11 The Script Editor window will show on screen, click to clear all button for clear old script and then press and hold shift button on keyboard and select all of Rig control in model, then use curser mouse cover all script in Script Editor drag and drop to menu tap and choose MEL script (Figure 3.63).

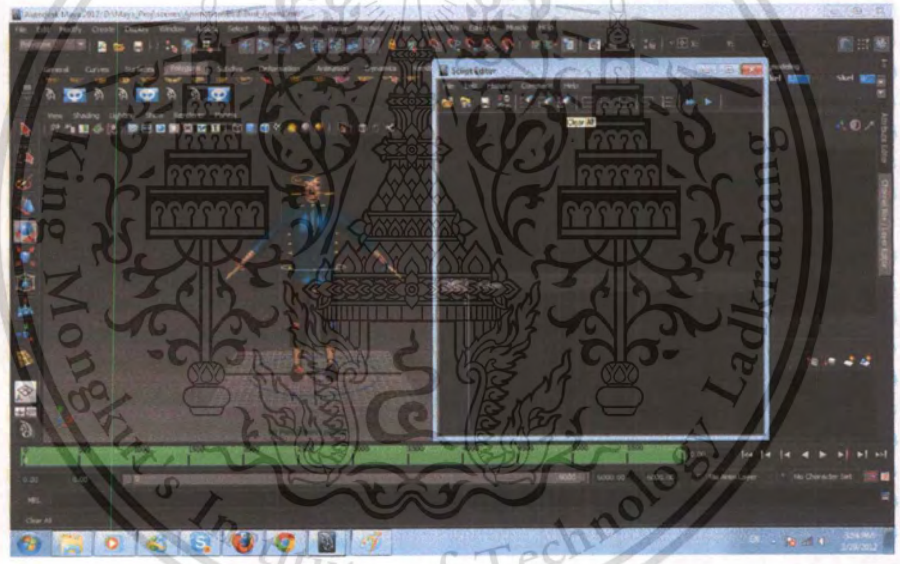


Figure 3.60: Script Editor window.

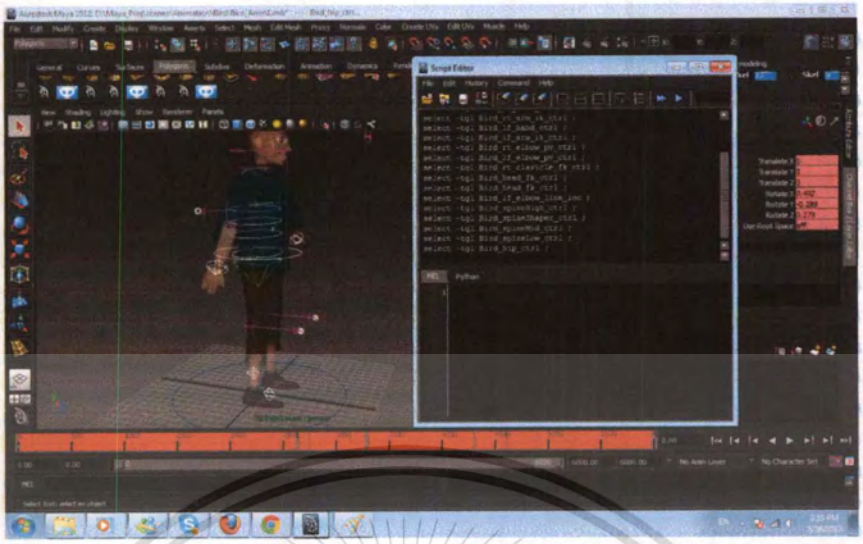


Figure 3.61: Choose Rig control to create MEL script.

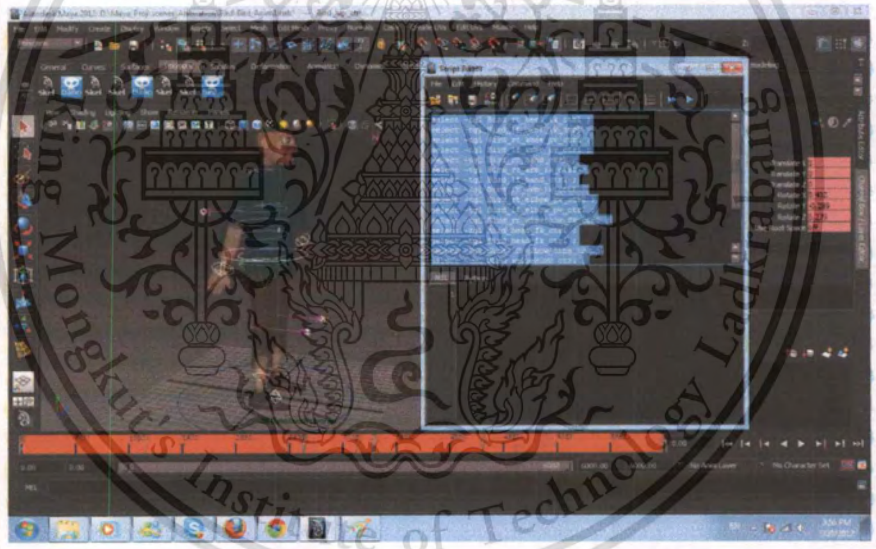


Figure 3.62: Select all of Rig control to create MEL script.

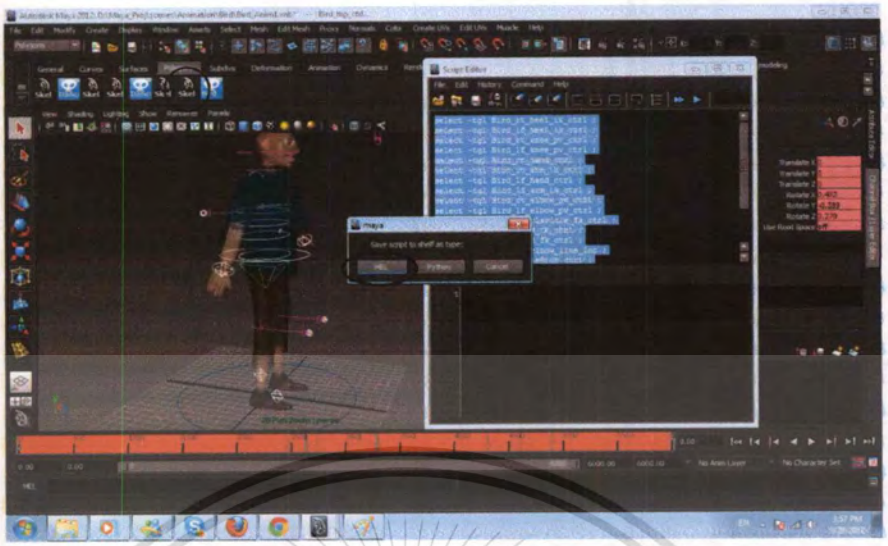


Figure 3.63: Drag and drop select script to menu tap and choose MEL script.

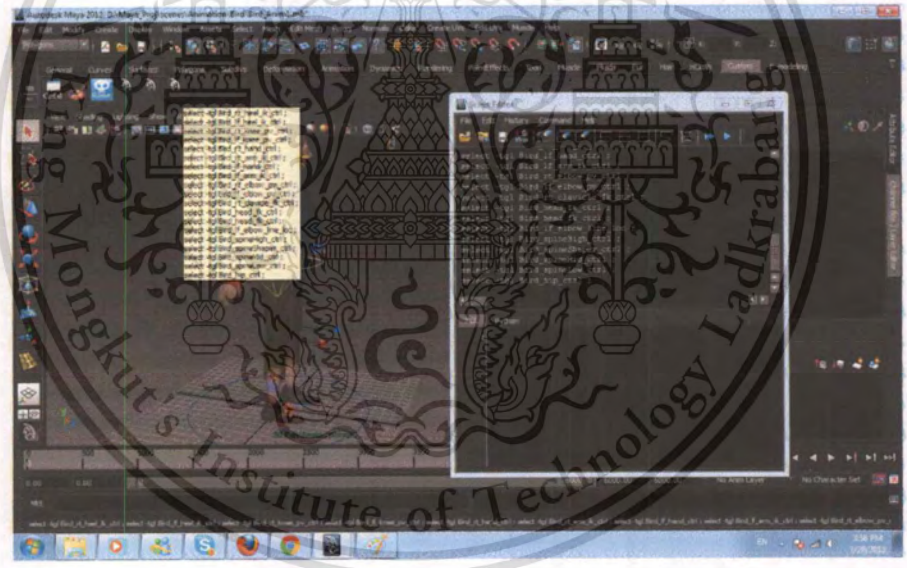


Figure 3.64: Control script in menu tap.

3.4.3.12 Assign the song to timeline for create animation, open sound folder drag and drop the song to timeline, the song format must be .wav format. Set end of time and playback range to 6000, 6000 and rang of song bar to 0.0, 0.0.

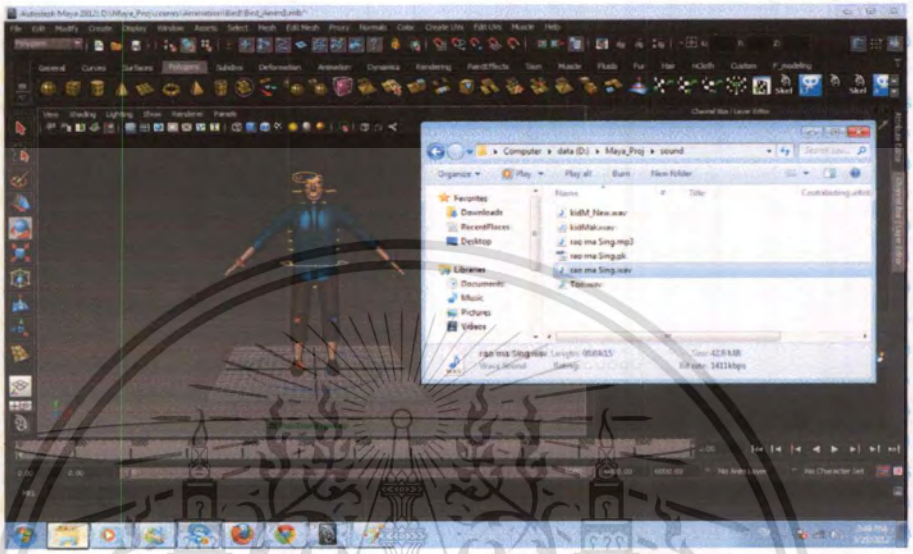


Figure 3.65: Drag and drop the song to timeline.

Figure 3.66: Set range of the song bar and set end of time and playback range.

3.4.3.13 Click Auto keyframe toggle (Figure 3.68) for use auto keyframe function. Create animation by click to rig control and use function Translate by press button W, Rotate by press button R and press S to adjust animation to that keyframe. Auto keyframe function will create animation when user move to other keyframe and adjust the movement of model (Figure 3.69).

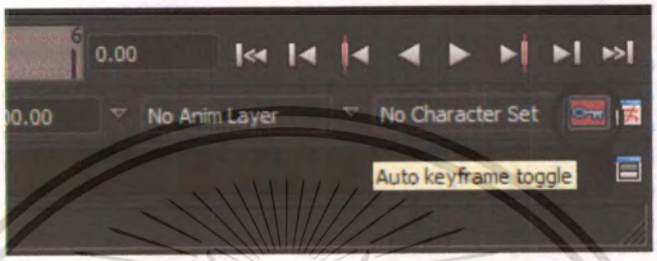


Figure 3.67: Auto keyframe toggle function.

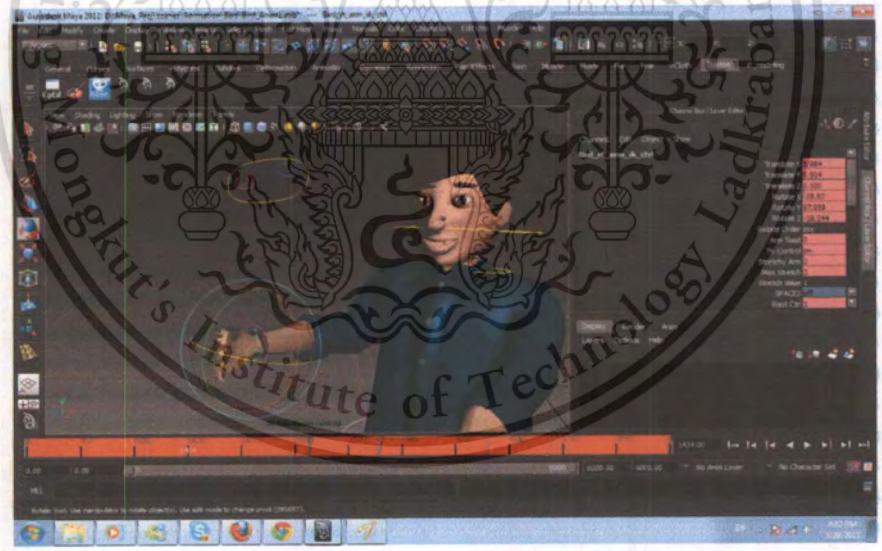


Figure 3.68: Create animation by Translate and Rotate the Rig control.

3.5 Set up the String™ library to Unity

3.5.1 String™ Library

This application the developer use the String™ Library to create the Augmented Reality on Iphone . The String™ is a high performance, Software Developer Kit (SDK) aimed at developers to create true augmented reality applications that can be experienced with an iOS device such as iPad or iPhone.

3.5.2 Basic requirement

- Unity 3 for iOS. With the Pro version (which allows editor plugins) on OS X and String™ preview functionality in the Unity editor.
- The latest Xcode and iOS SDK.
- A valid development provisioning profile.

3.5.3 Getting started

3.5.3.1 Open Unity.

Figure 3.69: Open Unity program.

3.5.3.2 In the Project Wizard, select the Create new Project tab.

Figure 3.70: Create new project.

3.5.3.3 Choose a folder and press Create Project.

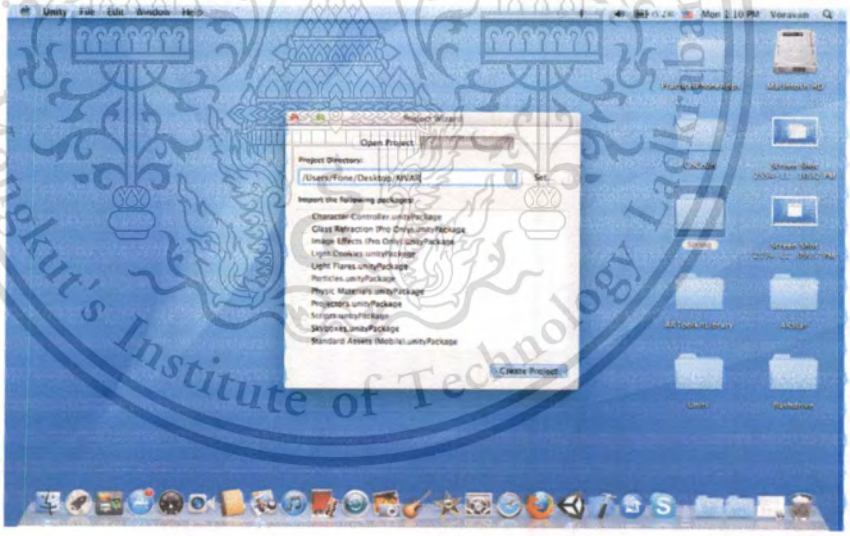


Figure 3.71: Create new project.

3.5.3.4 In the Project view, create folders Plugins, Scripts, StreamingAssets, Materials and Scenes. Please make sure the Plugins and StreamingAssets folders are spelled correctly and placed in your project root, as these folders have special meaning in Unity.

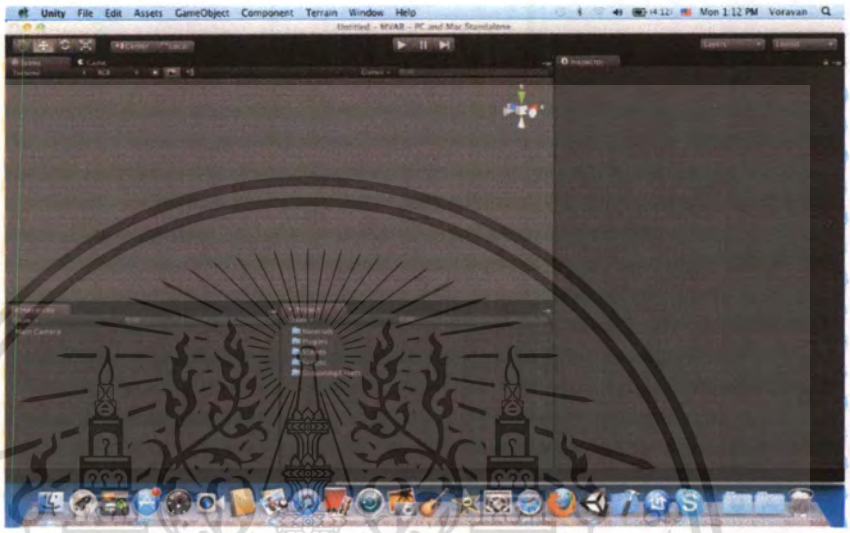


Figure 3.72: Create folders Plugins, Scripts, StreamingAssets, Materials and Scenes.

3.5.3.5 Drag and drop String.bundle from the Libraries folder in the String™ SDK to the Plugins folder in your project.

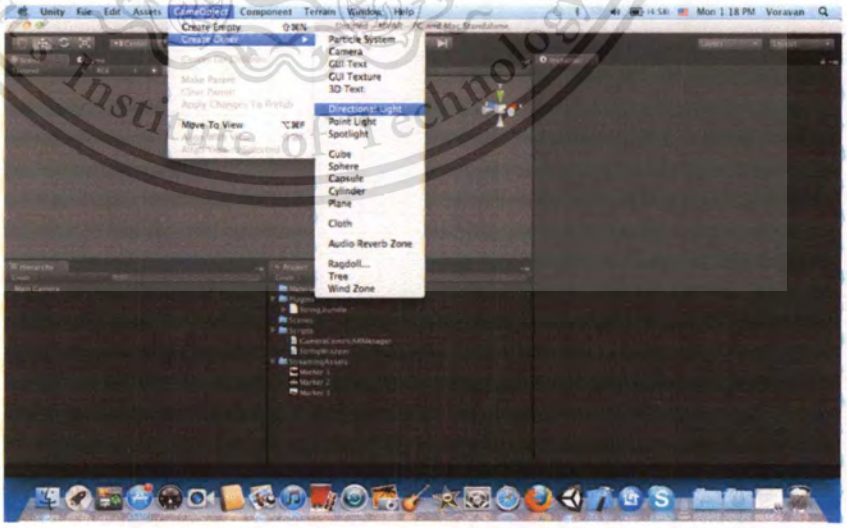


Figure 3.73: Drag and drop String.bundle to Unity.

3.5.3.6 Drag and drop StringWrapper.cs and CameraCentricARManager.cs from the Assets folder in the String™ SDK to the Scripts folder in your project.

Figure 3.74: Drag and drop StringWrapper.cs and CameraCentricARManager.cs to Unity.

3.5.3.7 Drag and drop Marker 1.png, Marker 2.png and Marker 3.png from the Assets folder in the StringTM SDK to the StreamingAssets folder in your project.

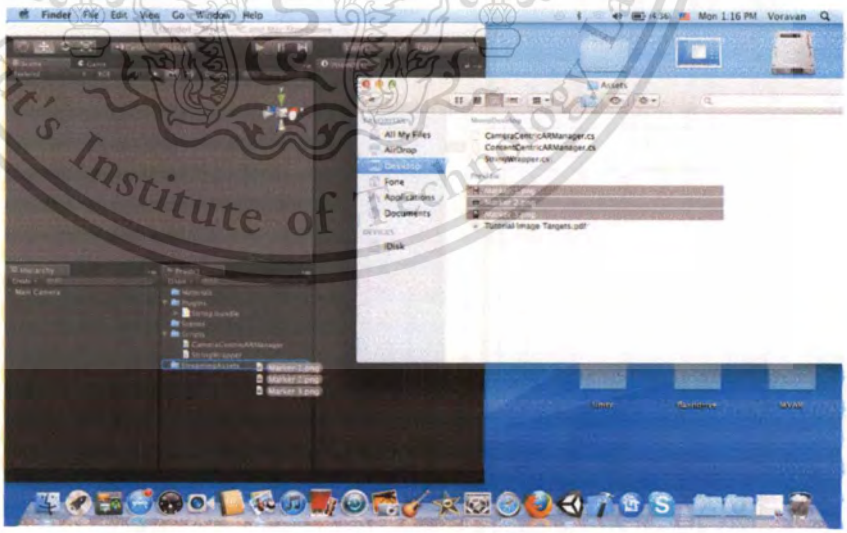


Figure 3.75: Drag and drop Marker to Unity.

3.5.4 Creating a directional light

3.5.4.1 In the Unity menu, select GameObject ⟶ Create Other ⟶ Directional Light.

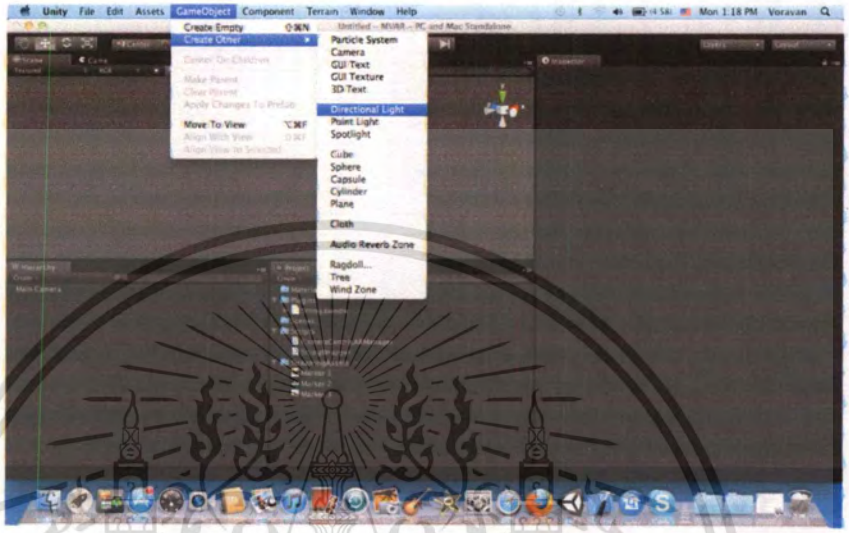


Figure 3.76: Create Directional Light.

3.5.4.2 Drag and drop the Directional Light onto Main Camera.

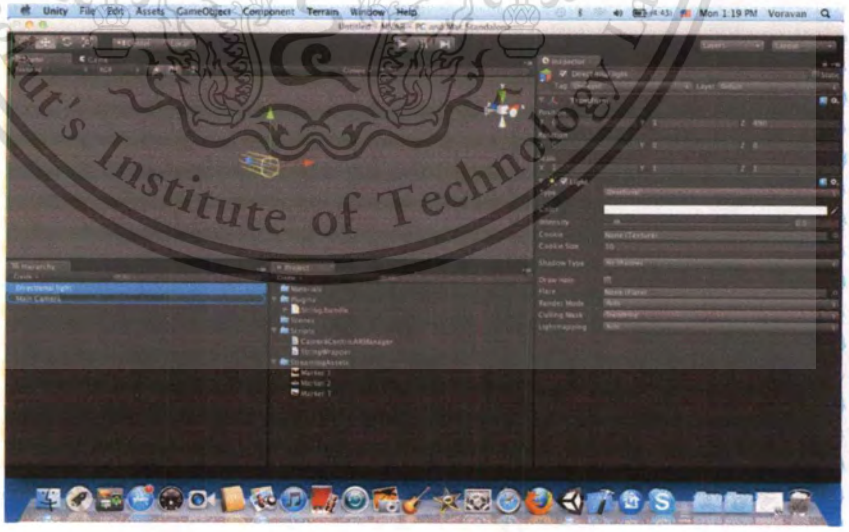


Figure 3.77: Drag and drop the Directional Light onto Main Camera.

3.5.5 Creating materials

3.6.5.1 Right-click the Materials folder in your Project view, and select Create ⟶Material.

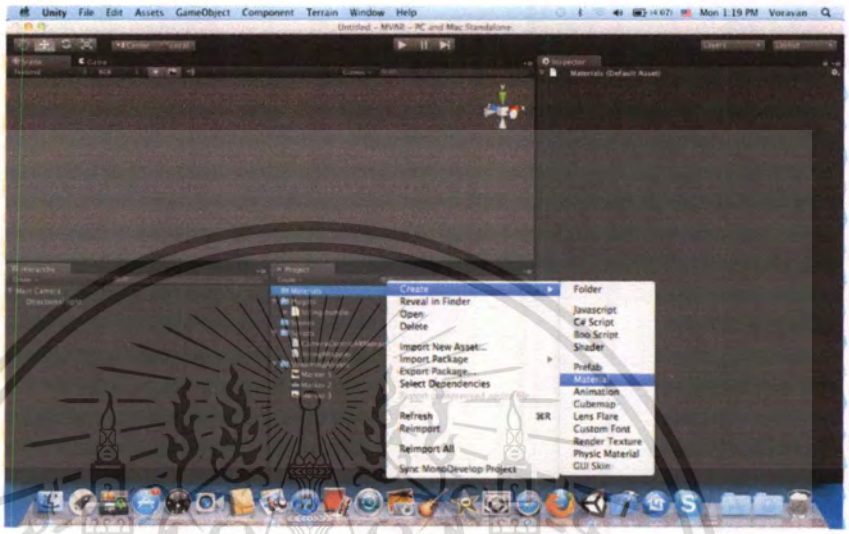


Figure 3.78: Create Materials.

3.5.5.2 Rename the new material Red. Click the material of Main Color and select a red color.

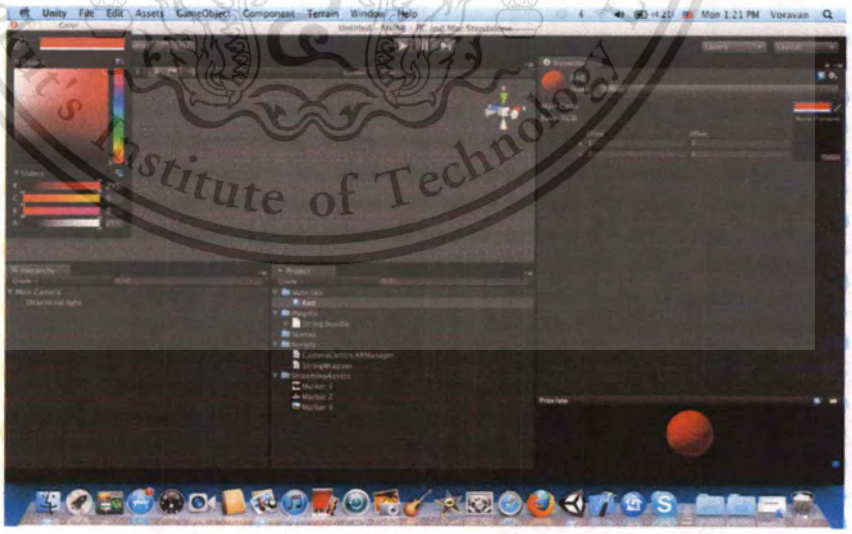


Figure 3.79: Rename the material.

3.5.5.3 Repeat the above two steps with blue and green.

3.5.6 Creating objects

3.5.6.1 In the Unity menu, select GameObject ⟶ Create Empty.

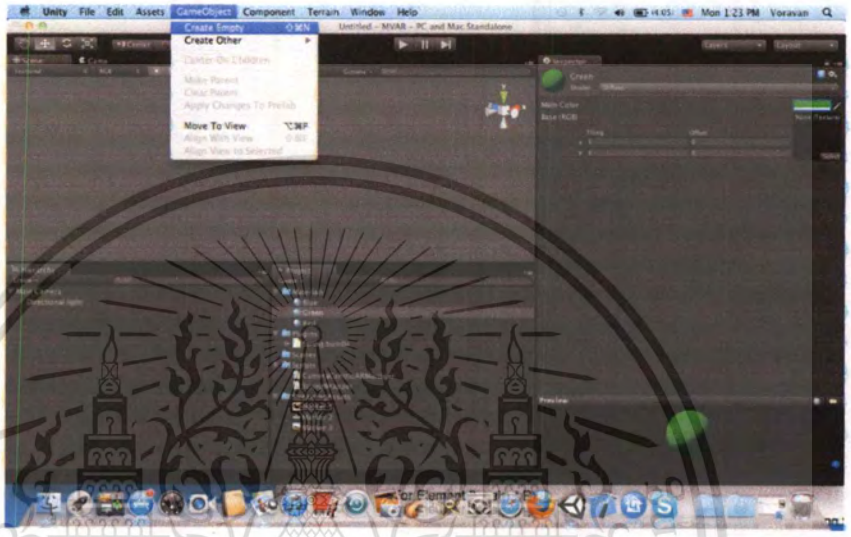


Figure 3.80: Create empty.

3.5.6.2 Rename the object to Root Object 1.

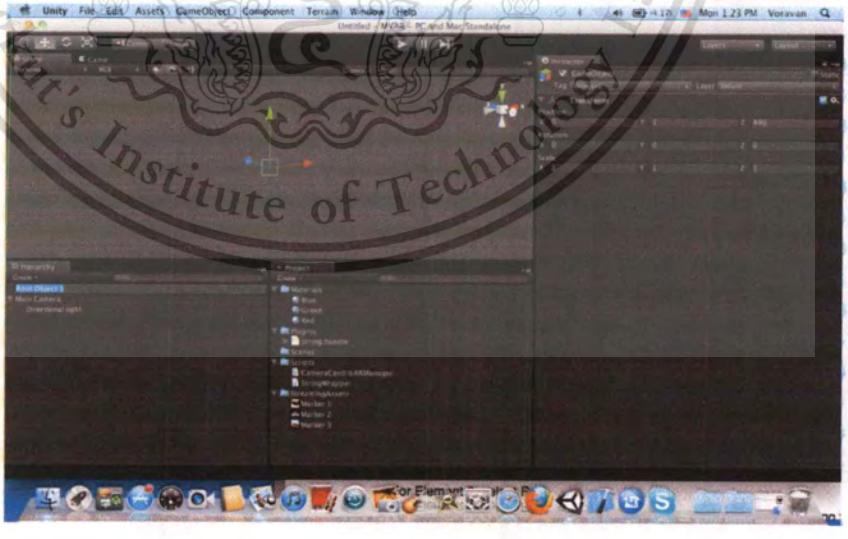


Figure 3.81: Rename the object to Root Object1.

3.5.6.3 Select GameObject ⟶ Create Other ⟶ Cube.

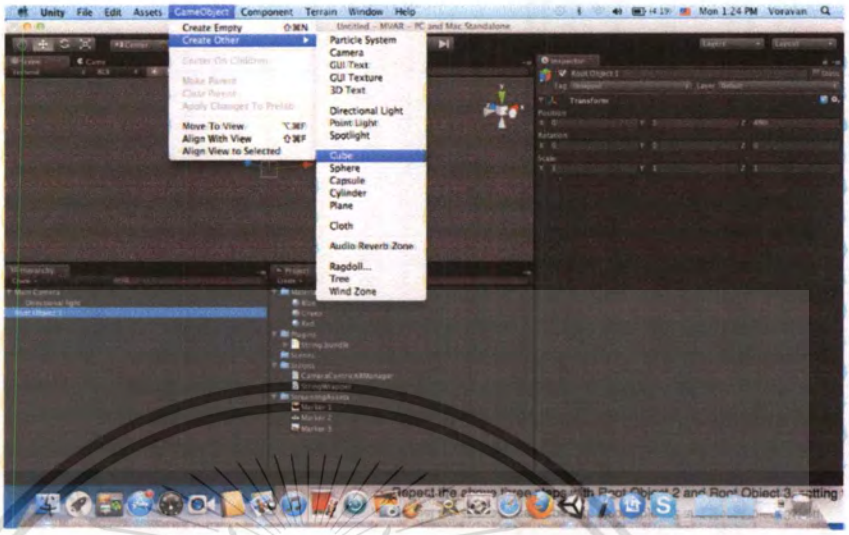


Figure 3.82: Create Cube.

3.5.6.4 Drag and drop Cube onto Root Object 1. Make sure rotation of Cube is 0, 0 and 0.

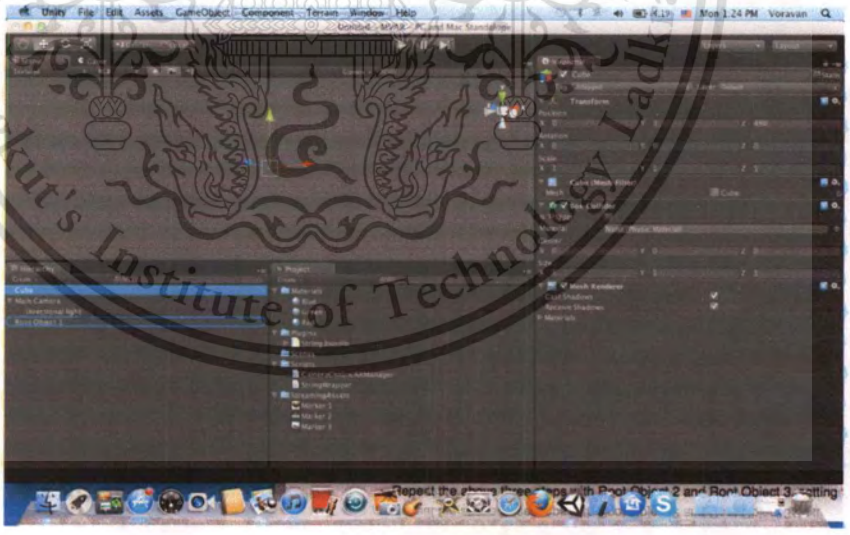


Figure 3.83: Drag and drop Cube onto Root Object 1.

3.5.6.5 Set the position to 0, 0 and 0.2.

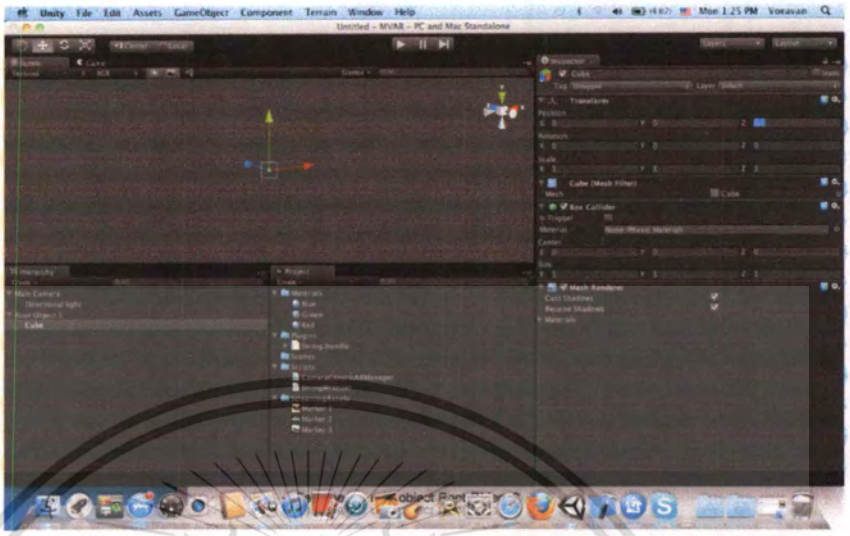


Figure 3.84: Set the position.

3.5.6.6 Set value of scale to 0.4, 0.4 and 0.4.

Figure 3.85: Set the scale.

3.5.6.7 Right-click Root Object 1 and select Duplicate.

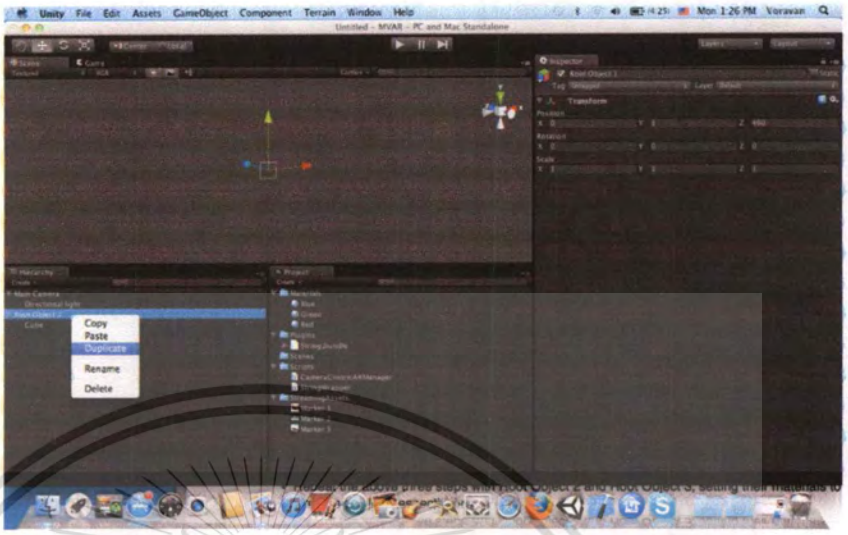


Figure 3.86: Duplicate the Root Object1.

3.5.6.8 Rename the new object Root Object 2.

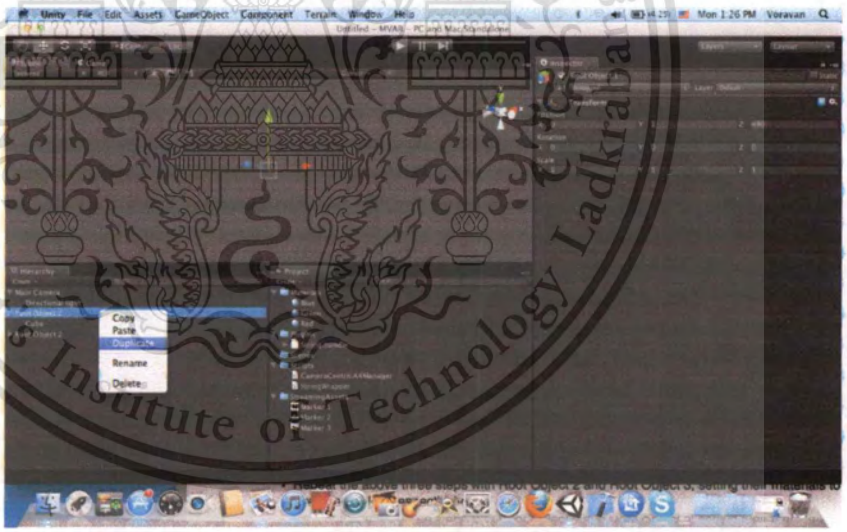


Figure 3.87: Rename new Root Object.

3.5.6.9 Duplicate Root Object 1 again, and rename the new object to Root Object 3.

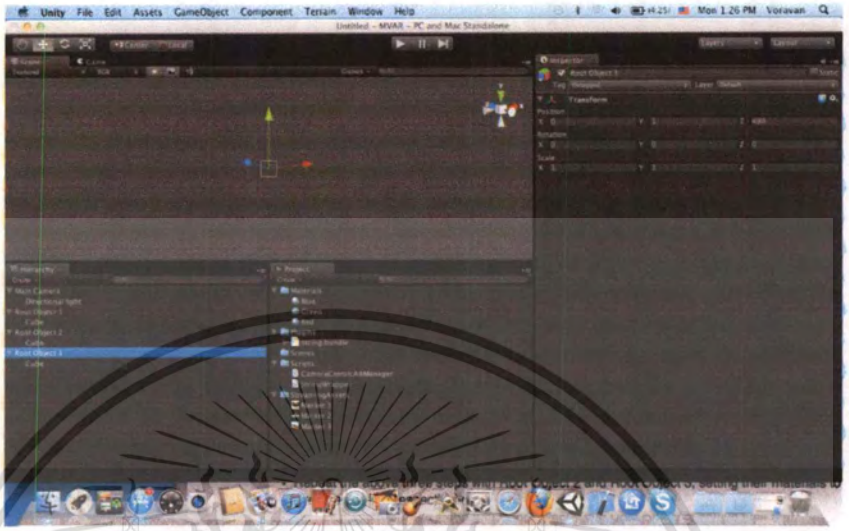


Figure 3.88: Duplicate the Root Object1 again.

3.5.6.10 Select the Cube attached to Root Object 1.

Figure 3.89: Select the Cube attached to Root Object 1.

3.5.6.11 In the Inspector view expand the Mesh Renderer Materials array. For Element 0, select Red.

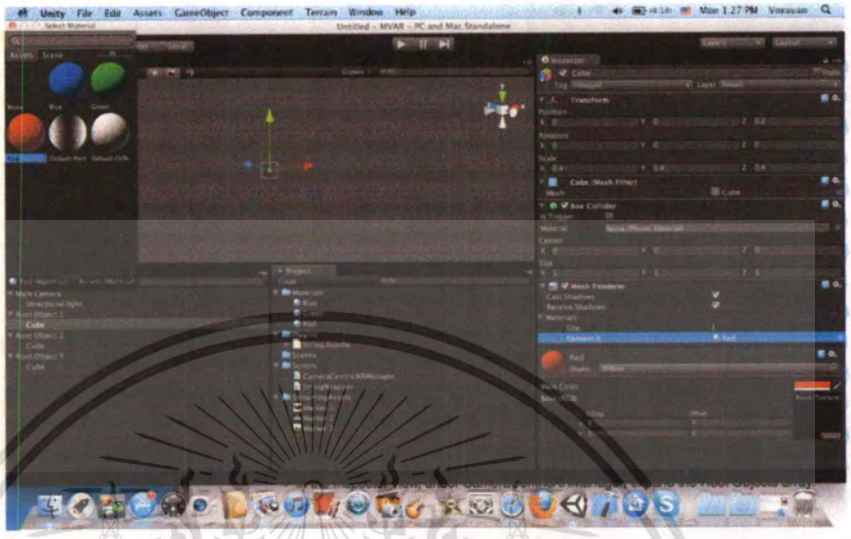


Figure 3.90: Expand the Mesh Renderer of Materials array.

3.6.6.12 Repeat the above three steps with Root Object 2 and Root Object 3, setting their materials to Green and Blue, respectively.

3.5.7 Setting up CameraCentricARManager

3.5.7.1 Drag and drop the CameraCentricARManager script onto Main Camera.

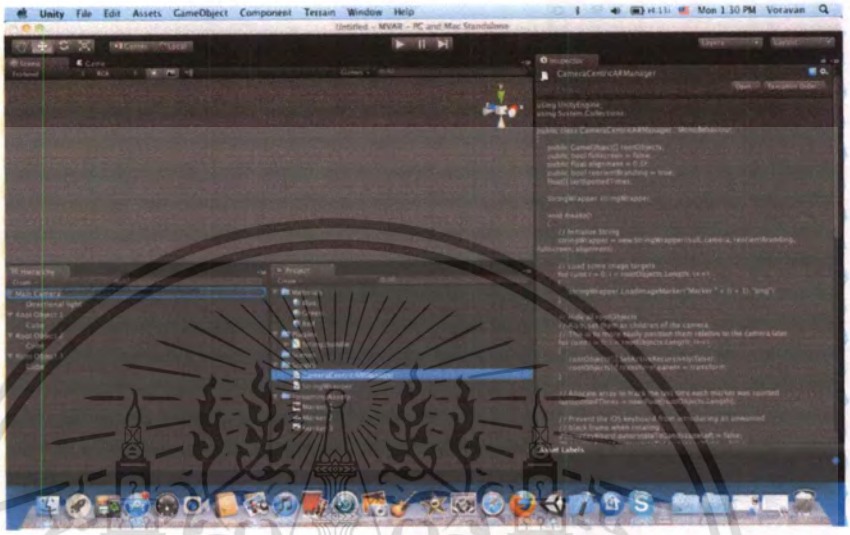


Figure 3.91: Drag and drop the CameraCentricARManager script onto Main Camera.

3.5.7.2 Select Main Camera.

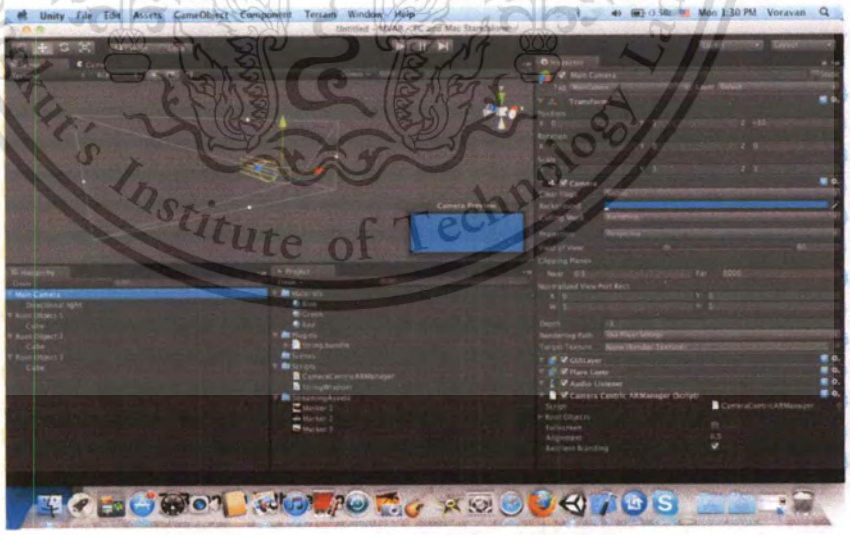


Figure 3.92: Select Main Camera.

3.5.7.3 In the Inspector view, under CameraCentricARManager, expand the Root Objects array.

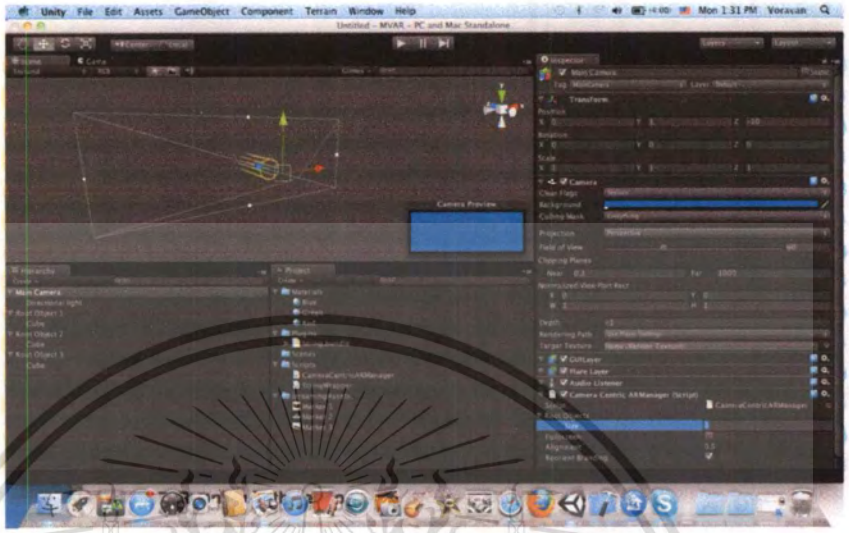


Figure 3.93: Expand the Root Objects array.

3.5.7.4 Set its size to 3.

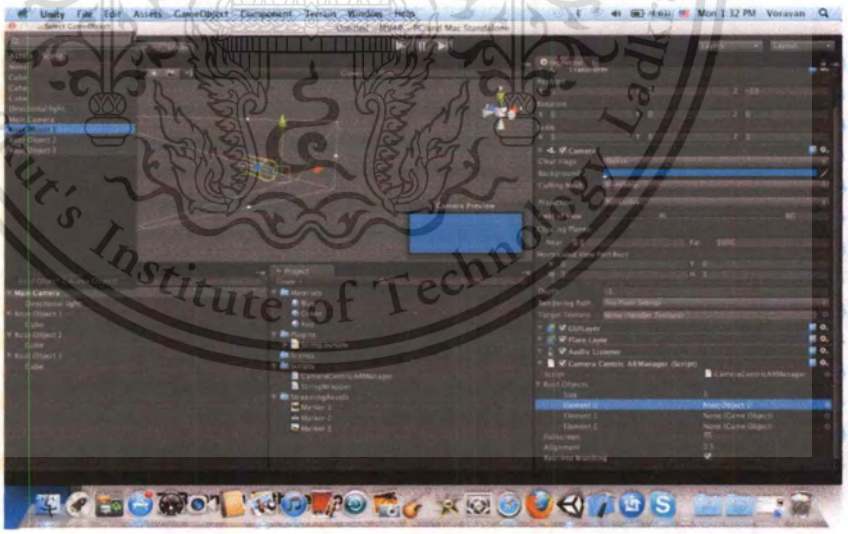


Figure 3.94: Set size to 3.

3.5.7.5 Set its elements to Root Object 1, 2 and 3.

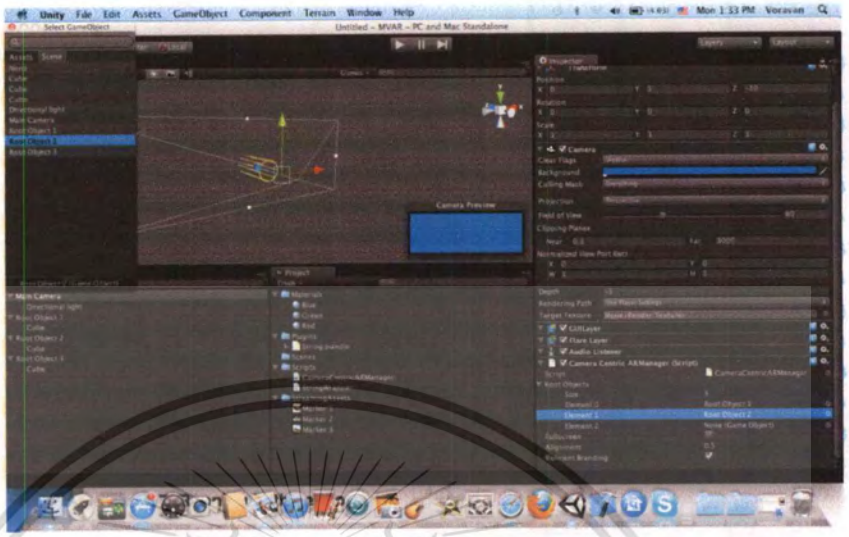


Figure 3.95: Set its elements to Root Object 1, 2 and 3.

3.5.7.6 Press cmd+S to save the scene. Save it as Main in the Scenes folder.

Figure 3.96: Save the scene.

3.5.8 Running in the editor

3.5.8.1 Select File ⟶ Build Settings.

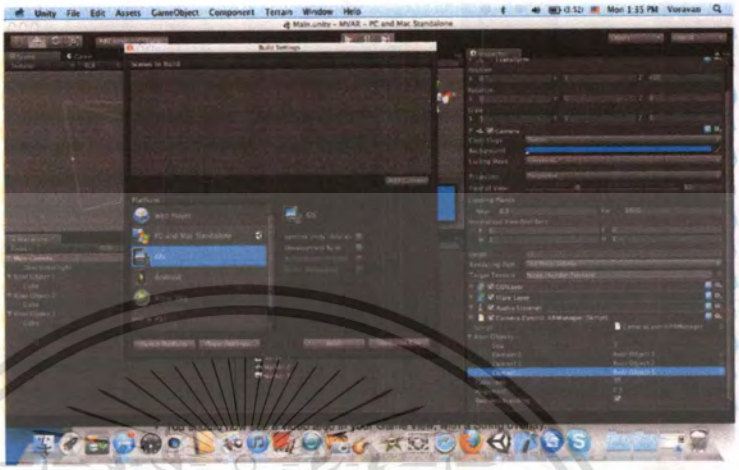


Figure 3.97: Build setting.

3.5.8.2 Select iOS and press Switch Platform. Close the Build Settings window. Print out one or more of the image markers. And hold the printed image marker in front of the camera.

Figure 3.98: Switch the platform.

3.6 Create the application in Unity

3.6.1 Delete three old markers in the Streaming Assets, Root Object 2, Root Object 3, which is created from setting up the String™ Library by press command and delete.

3.6.2 Drag and drop Bird Marker to the Streaming Assets, which is in the Project window.

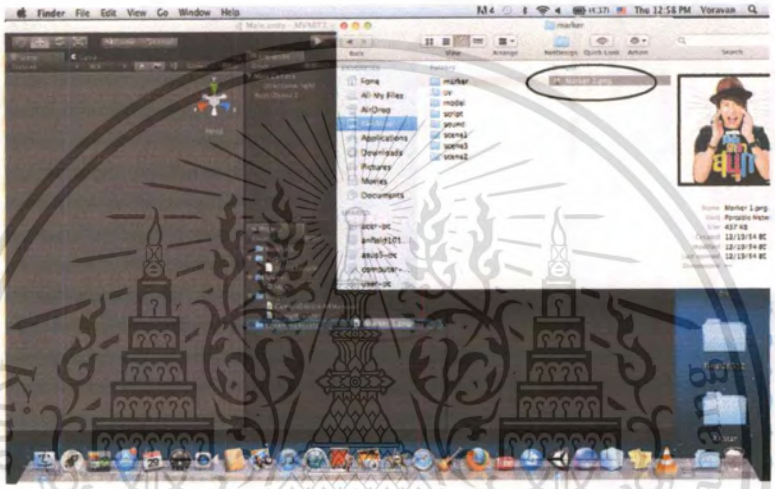


Figure 3.99: Drag and drop marker to Project window.

3.6.3 Bird Marker will be in the Streaming Assets folder.

Figure 3.100: Streaming Assets folder.

3.6.4 Drag and drop the .fbx model, which export from Autodesk Maya to Materials folder. The model uses in function of changing the cloth, it does not have an animation.

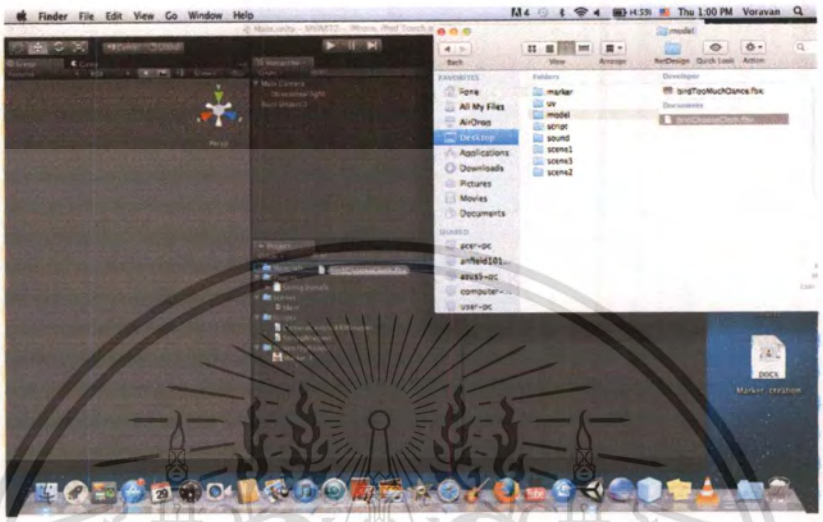


Figure 3.101: Drag and drop .fbx model to Project window.

3.6.5 The file will be in the Materials folder.

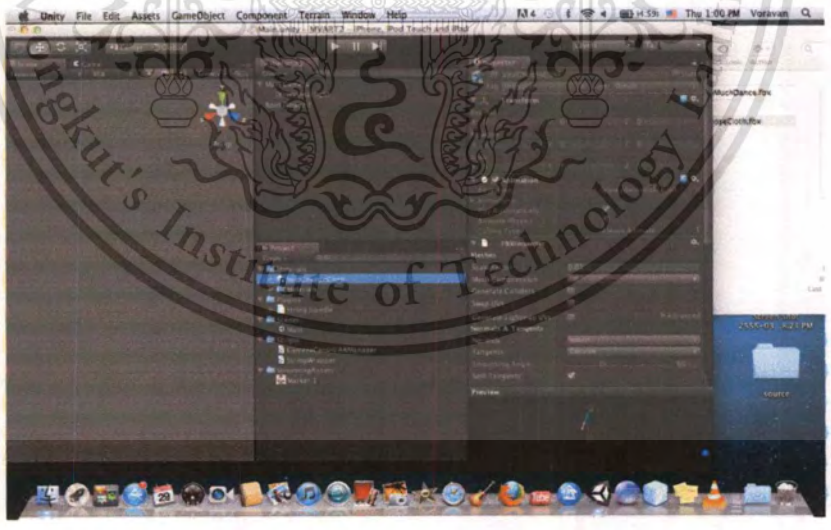
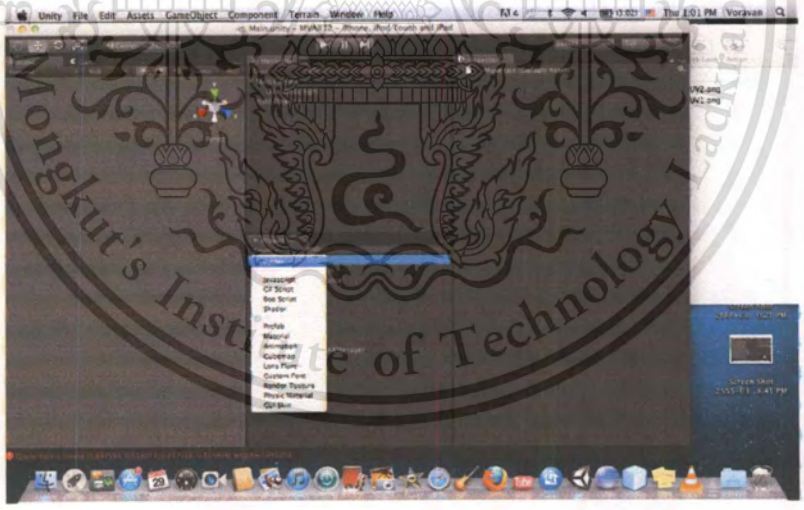


Figure 3.102: Materials folder.

3.6.6 Drag and drop the .fbx animation model, which export from Autodesk Maya to Materials folder. The model uses in dance scene.

Figure 3.103: Drag and drop the .fbx animation model toMaterials folder.



3.6.7 Create the new folder.

3.6.8 Rename the new folder to UVtexture.

Figure 3.105: Rename the folder to UVtexture.

3.6.9 There are two different cloths of UV texture. The first one calls birdOutUV1.

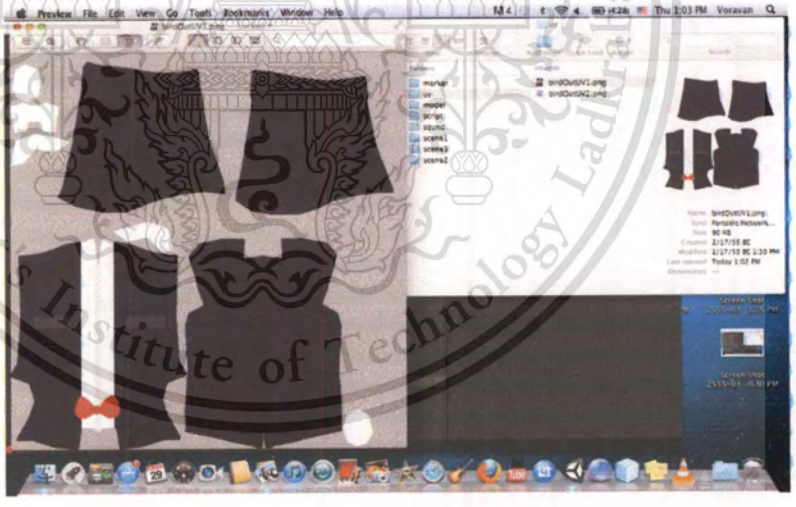


Figure 3.106: Two UV texture.

3.6.10 The second one calls birdOutUV2.

Figure 3.107: birdOutUV2.

3.6.11 Drag and drop two UV texture to UVtexture folder.

Figure 3.108: Drag and drop two texture to UVtexture folder.

3.6.12 Two UV texture will be in the UV texture folder.

Figure 3.109: UV texture folder.

3.6.13 Create new folder.

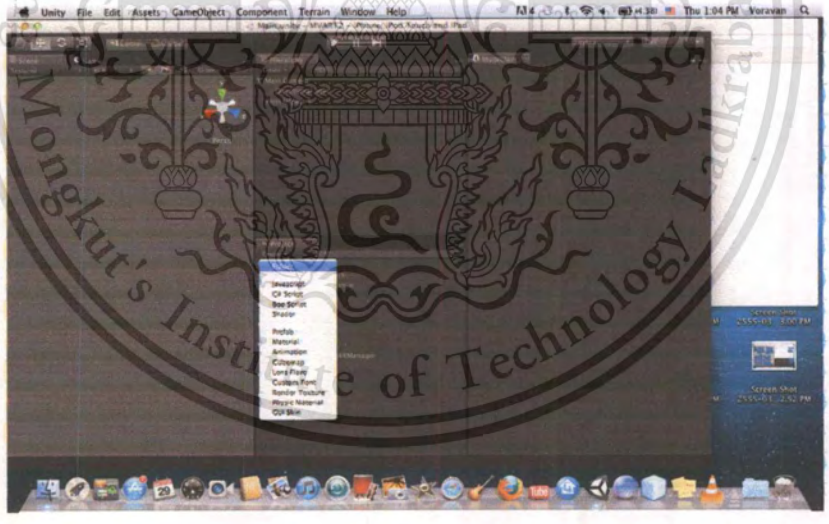


Figure 3.110: Create new folder.

3.6.14 Rename the new folder to UI.

Figure 3.111: UV texture folder.

3.6.15 Drag and drop to the folders scene1, scene2, and scene3. Each folder collects the user's interface of the application and divided it to the subfolder, one folder per one scene. The user's interface of each application can create it from Adobe Illustrator or Adobe Photoshop program.

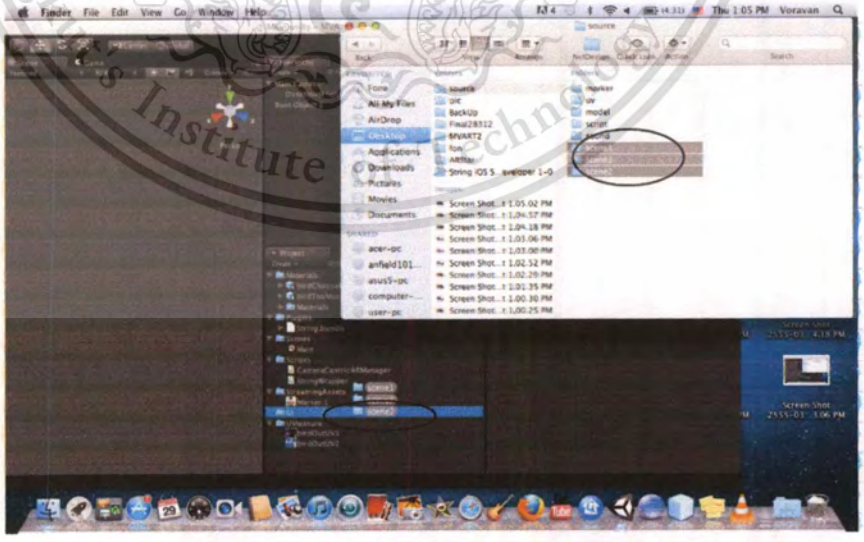


Figure 3.112: Drag and drop the folder to Project window.

3.6.16 The UI of three scenes will be in the UI folder.

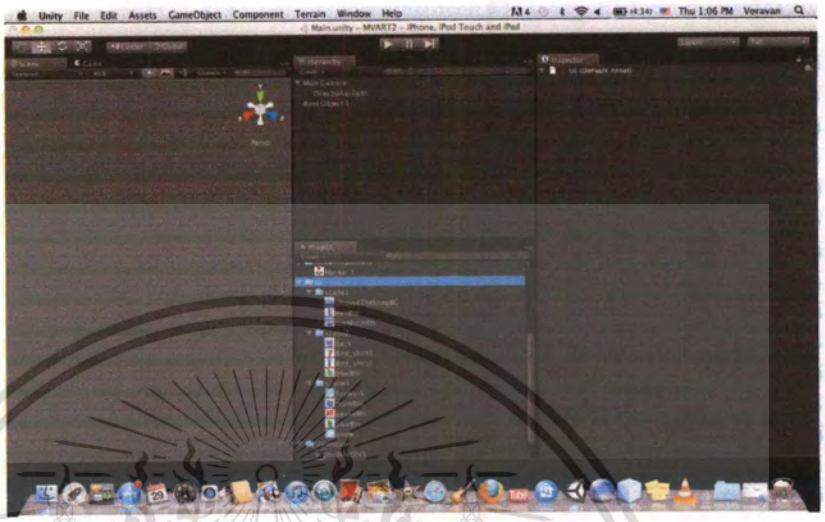


Figure 3.113: UI folder.

3.6.17 Change type of the texture from each user texture interface to GUI.

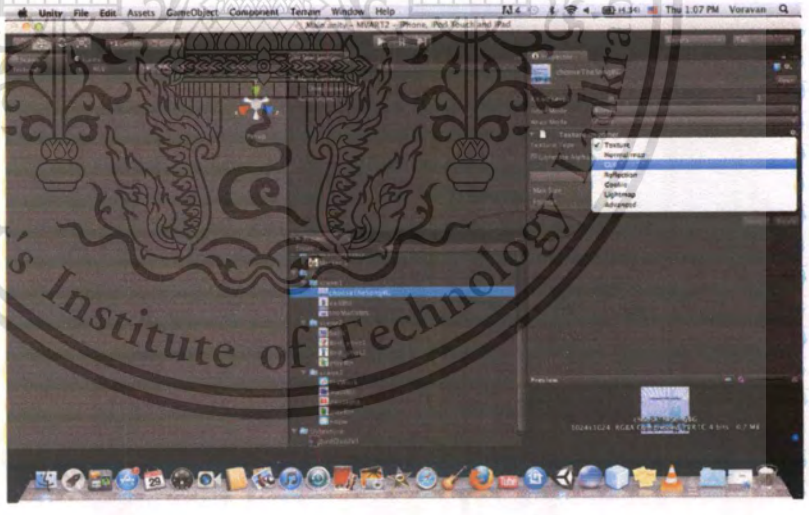


Figure 3.114: Change type of texture.

3.6.18 After changed the texture type to GUI, press Apply button.

Figure 3.115: Press Apply button.

3.6.19 Select the chooseTheSongBG, which is in the UI folder. Select GameObject>Create Other>GUI Texture.

Figure 3.116: Create GUI texture.

3.6.20 Select the ChooseTheSongBG, which is in the Hierarchy window and set Width to 640 and Height to 960.

Figure 3.117: Set the texture position.

3.6.21 Right click at the Script folder and choose Create>Javascript file.

Figure 3.118: Create Java script.

3.6.22 Rename the file to changeSizeScreen.

Figure 3.119: Rename the script.

3.6.23 Code the Java Script language to fix to IPhone screen size.

```
function Start ()

{

        guiTexture.pixelInset.width = Screen.width;

        guiTexture.pixelInset.height = Screen.height;

}
```

3.6.24 Select the ChooseTheSongBG, which is in the Hierarchy view and drag and drop changeSizeSceen script to empty space in Inspector view.

Figure 3.120: Drag and drop changeSizeSceen script to Inspector view.

3.6.25 The changeSizeSceen script will be import to ChooseTheSongBG.

Figure 3.121: Import the script to texture.

3.6.26 Select tooMuchBtn texture, which is in Project view.

Figure 3.122: Select the texture.

3.6.27 Select GameObject>Create Other>GUI Texture

Figure 3.123: Create GUI texture.

3.6.28 Create Java Script file.

Figure 3.124: Create Java Script file.

3.6.29 Rename the file to changeScene.

Figure 3.125: Rename the script.

3.6.30 Write the code to change the scene using Java Script language.

```
var levelToLoad : String;

var normalTexture : Texture2D;

var rollOverTexture : Texture2D;

var beep : AudioClip;


//Quit

var quitButton : boolean = false;


function OnMouseEnter ()

{

        guiTexture.texture = rollOverTexture;

}

function OnMouseExit()

{

        guiTexture.texture = normalTexture;
```

```
}

function OnMouseUp()

{

        audio.PlayOneShot(beep);

        yield new WaitForSeconds(0.35); // "wait" until new thread


        if(quitButton)

        {

            Application.Quit();

        }

        else

        {

            Application.LoadLevel(levelToLoad);

        }

}


function Update()

{

        for (var touch : Touch in Input.touches)
```

```
{
        if(guiTexture.HitTest(touch.position) && touch.phase ==
TouchPhase.Ended )
                {
                        audio.PlayOneShot(beep);
                        if(quitButton)
                        {
                                Application.Quit();
                        }
                        else
                        {
                                Application.LoadLevel(levelToLoad);
                        }
                }
        }
}
@script RequireComponent(AudioSource)
```

3.6.31 Select the tooMucBtn in the Hierarchy view, drag and drop changeScene script into empty space.

Figure 3.126: Drag and drop the script into empty space.

3.6.32 The tooMuchBtn will has changeScene script.

Figure 3.127: ChangeScene script will be imported to texture.

3.6.33 Drag and drop the audio clip that wanted which will use in the application. In this sample application, use for Too much so much very much song, and beep sound will active when the user touches the button.

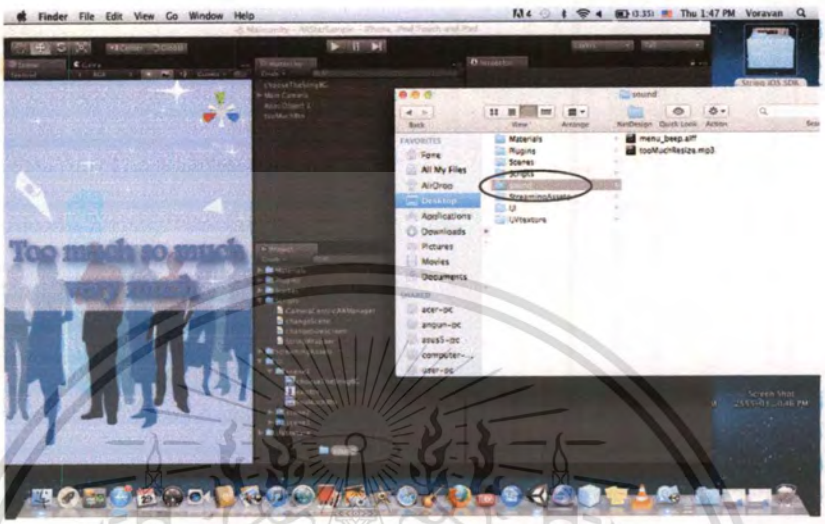


Figure 3.128: Drag and drop the audio clip to Project window.

3.6.34 The audio clip will be in the Project view.

Figure 3.129: Sound folder.

3.6.35 Select tooMuchBtn and drag and drop menu_beep audio clip to Beep in change Scene script, in the Inspector view.

Figure 3.130: Drag and drop the audio clip to change Scene script.

3.6.36 Drag and drop tooMuchBtn texture to Normal texture and Roll over texture in the change Scene script, in Inspector view.

Figure 3.131: Drag and drop the texture to change Scene script.

3.6.37 Press command and d to duplicate the Main scene.

Figure 3.132: Duplicate the Main scene.

3.6.38 Rename the scene to chooseCloth.

Figure 3.133: Rename the scene.

3.6.39 Open the chooseCloth scene. Select exitBtn texture and select Game Other>Create Other>GUI Texture.

Figure 3.134: Create GUI texture.

3.6.40 The exit texture button will be on the screen

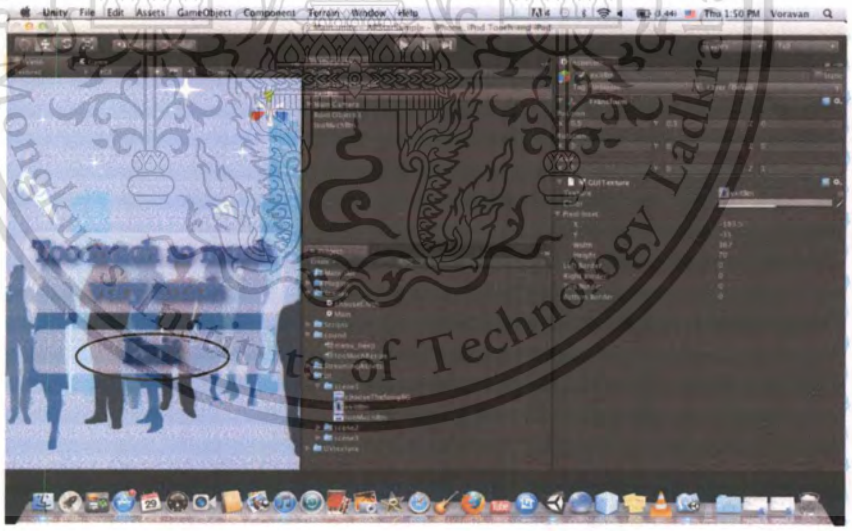


Figure 3.135: Exit texture button.

3.6.41 Select exitBtn texture button and drag and drop changeScene script to the empty space in Inspector view.

Figure 3.136: Drag and drop script to the in Inspector view.

3.6.42 Drag and drop menu beep audio clip, exitBtn texture to changeScene script in the Inspector view.

Figure 3.137: Drag and drop audio clip, button texture to script in the Inspector view.

3.6.43 Delete Directional light from Main Camera by pressing command and delete.

Figure 3.138: Delete Directional light.

3.6.44 This scene does not have to use the Directional light.

Figure 3.139: Delete Directional light.

3.6.45 Open chooseCloth scene and drag and drop birdChooseCloth.fbx from Project view to Root Object 1, which is in the Hierarchy view.

Figure 3.140: Drag and drop the model from Project view to Root Object 1.

3.6.46 The birdChooseCloth.fbx will be in the Root Object 1.

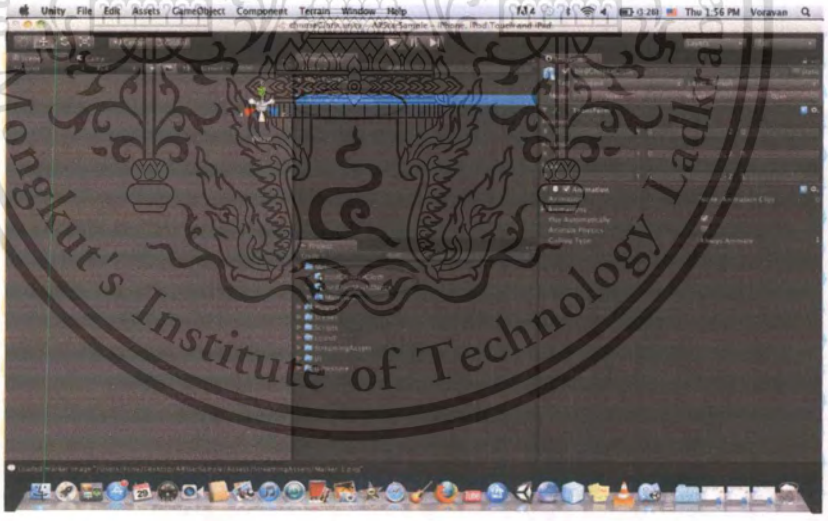


Figure 3.141: Root Object 1.

3.6.47 Select Bird_shirt1 from UI folder, in the Project view. Select GameObject>Create Other>GUI Texture to create first change cloth button. The Bird_shirt1 button will be appear on the screen. Set the position from the Inspector view.

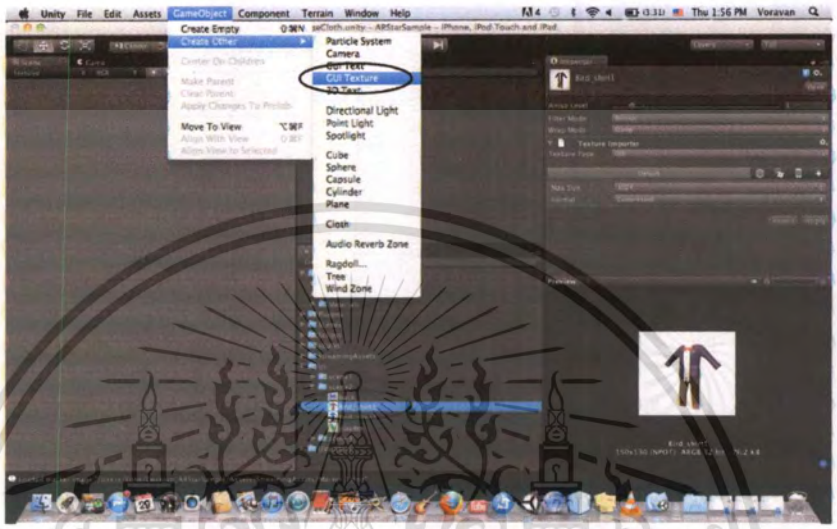


Figure 3.142: Create GUI texture.

3.6.48 Right click at Scripts folder and create Java Script file.

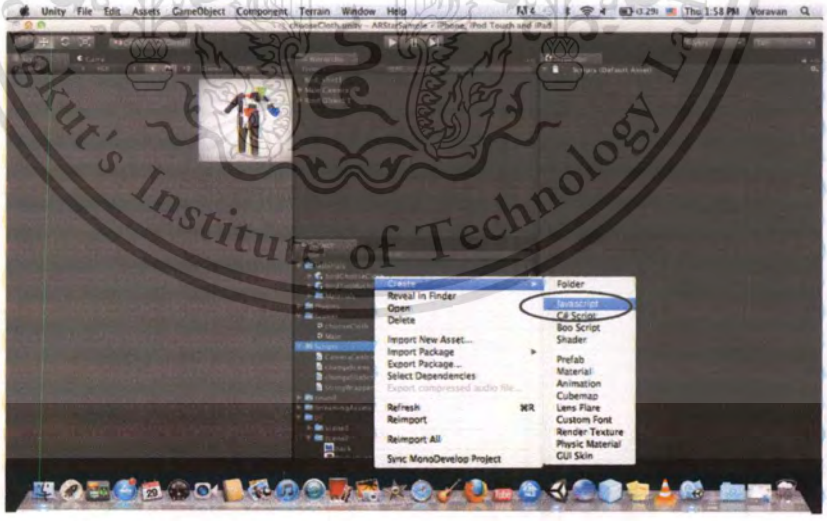


Figure 3.143: Create Java Script file.

3.6.49 Rename the script to changeCloth.

Figure 3.144: Rename the script.

3.6.50 Open the changeCloth script and write the java script code.

```
var textTure : Texture2D;
var target : GameObject;

function OnMouseUp()
{
        target.renderer.material.mainTexture = textTure;
        Global.currentCloth = textTure;
}
```

```
function Update()

{

        for (var touch : Touch in Input.touches)

        {

            if(guiTexture.HitTest(touch.position) && touch.phase ==
TouchPhase.Ended )

            {

                target.renderer.material.mainTexture = textTure;

                Global.currentCloth = textTure;

            }

        }

}
```

3.6.51 Right click at Scripts folder and create Java Script file.

Figure 3.145: Create Java Script file.

3.6.52 Rename the script to Global.

Figure 3.146: Rename the script.

3.6.53 Open the Global script, write the java script code. Global script used for recording the action of clothing when the user chooses the singer's cloth.

```
//record

static var currentCloth : Texture2D;

static var currentSong : String;



function Update () {

}
```

3.6.54 Select Bird_shirt1 button, which is in the Hierarchy view. Drag and drop chooseCloth script to empty space in the Inspector view.

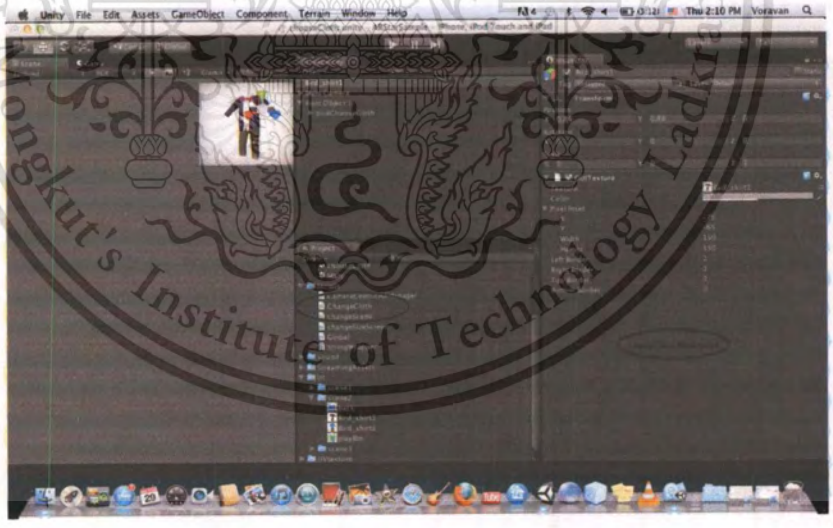


Figure 3.147: Drag and drop the script to empty space in the Inspector view.

3.6.55 Expand birdChooseCloth file in the Project view and find the UV object. When select the object from Project view, it will show picture of model beside. It is easily for the user to find the object that they wanted. In this simple AR application for IPhone, the UV object is only shirt. When change cloth, the bird model will change only the shirt color.

Select Bird_shirt1 button in the Hierarchy view. Drag and drop the UV object from birdChooseCloth file in the Hierarchy view to Target in changeCloth script, which belong to birdChooseCloth file in the Hierarchy.

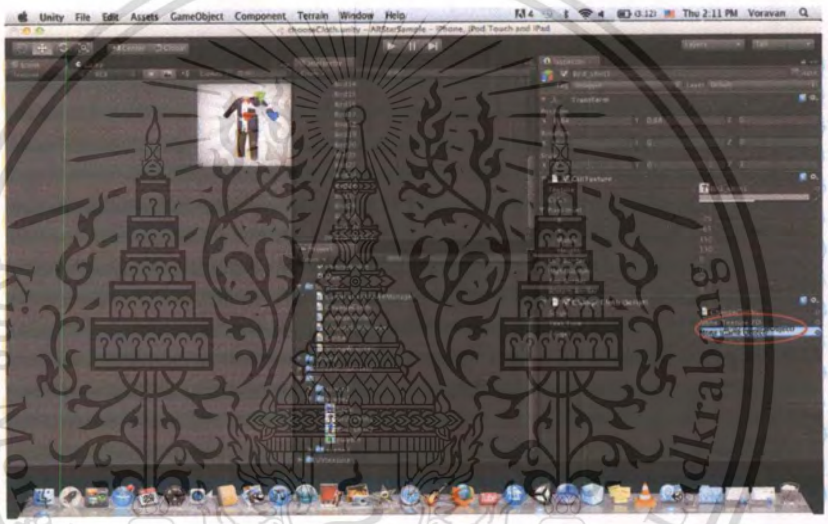


Figure 3.148: Drag and drop the UV object from model file in the Hierarchy view to Target in changeCloth script.

3.6.56 Drag and drop Bird_shirt1.png button texture to the texture field in the Inspector view.

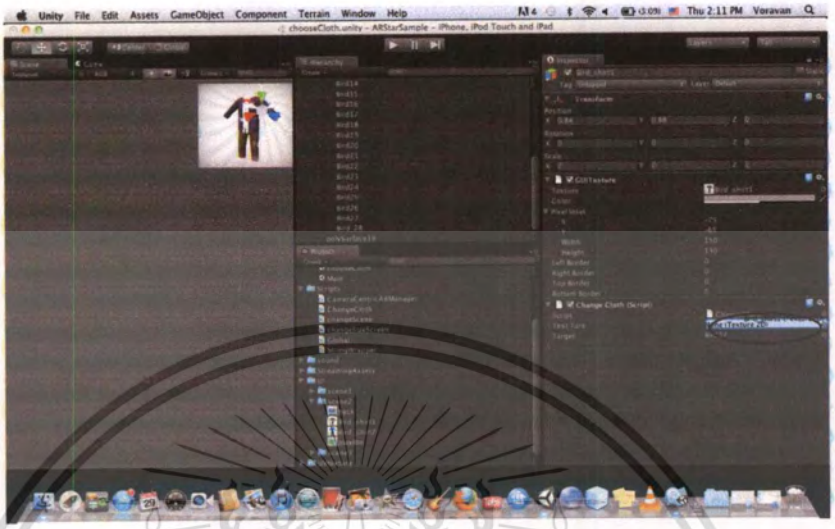


Figure 3.149: Drag and drop button texture to the texture field in the Inspector view.

3.6.57 Create second cloth button by repeating the previous step.

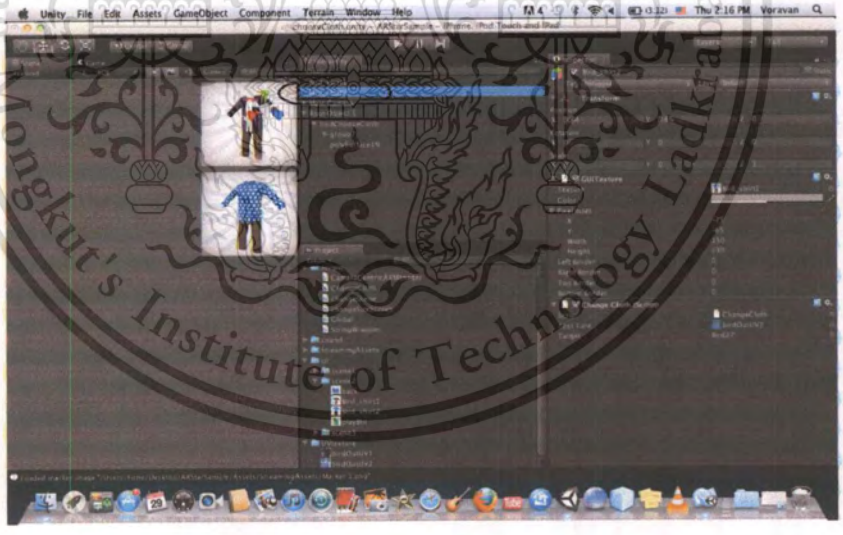


Figure 3.150: Second cloth button.

3.6.58 Select on UI folder the Back.png texture, which is in the Project view.

Figure 3.151: Select on UI folder the Back.png texture.

3.6.59 Select GameObject>,Create Other>GUI Texture to create Back button.

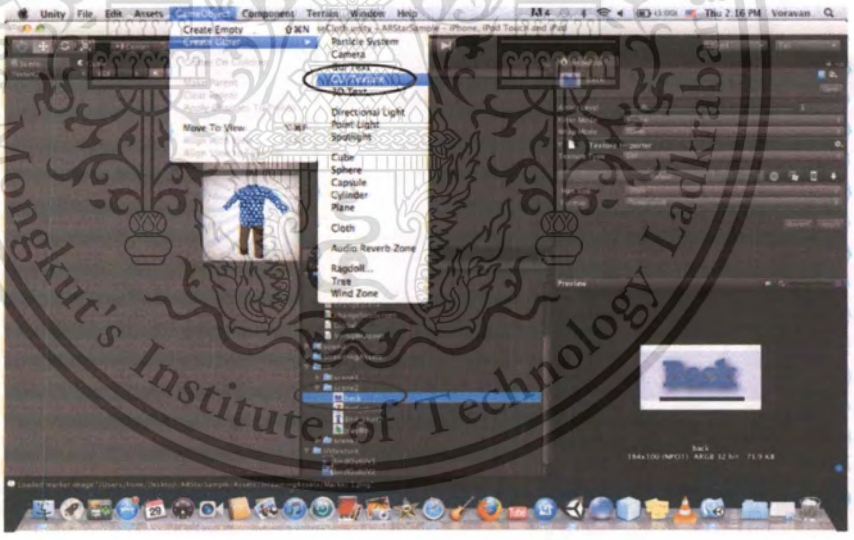


Figure 3.152: Create GUI texture.

3.6.60 Arrange the button position.

Figure 3.153: Arrange the position.

3.6.61 Select Play.png texture.

Figure 3.154: Select the button texture.

3.6.62 Select GameObject>Create Other>GUI Texture to create Play button.

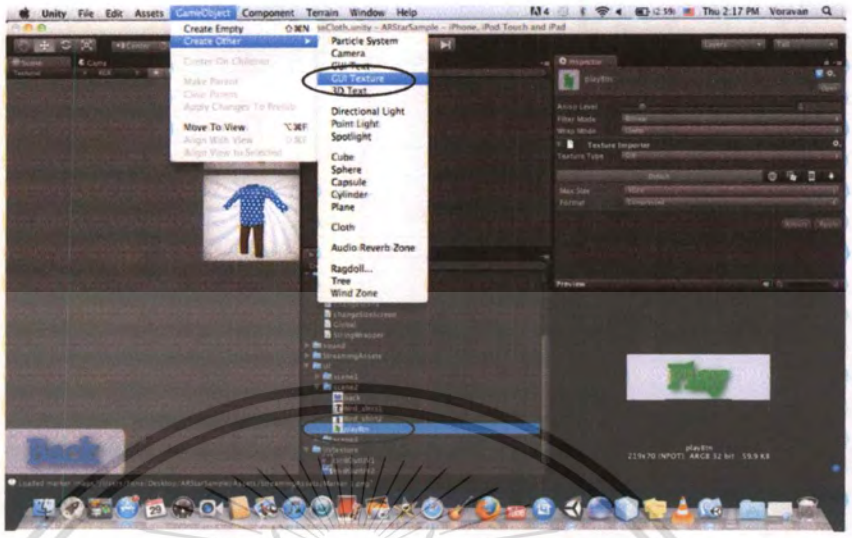


Figure 3.155: Create GUI texture.

3.6.63 Arrange the button position.

Figure 3.156: Arrange the position.

3.6.64 Select Back button in the Heirarchy view. Drag and drop the changeCloth.js from the Scripts folder, which is in the Project scene to the empty space in the Inspector view.

Figure 3.157: Drag and drop the script to the Inspector view.

3.6.65 Drag and drop Back.png texture from UI folder in Project view to Normal Texture field and Roll Over Texture field in the Inspector view and menu_beep.aiff audio clip from Sound folder in the Project view to Beep field in the Inspector view.

Figure 3.158: Drag and drop texture, audio clip to field in the Inspector view.

3.6.66 Type Main in Level To Load field, which is in the Inspector view.

Figure 3.159: Type the scene in Level To Load field.

3.6.67 Right click at Script folders in the project view to create Java Script file.

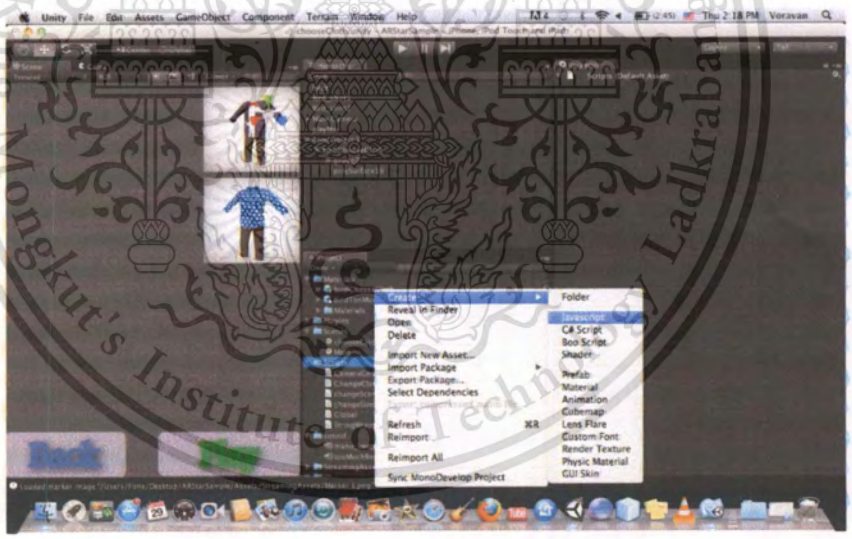


Figure 3.160: Create Java Script file.

3.6.68 Rename the Java file to play.

Figure 3.161: Rename the script.

3.6.69 Coding by using Java Script language in order to change the scene to birdDance scene. Drag and drop the script to playBtn button, which is in the Hierarchy view.

```
function OnMouseUp()
{
        Application.LoadLevel("birdDance");
}
function Update()
{
        for (var touch : Touch in Input.touches)
```

```
{

        if(guiTexture.HitTest(touch.position) && touch.phase ==
TouchPhase.Ended )

        {

                Application.LoadLevel("birdDance"); //
birdDance=Scene name

        }

}

}
```

3.6.70 Select chooseCloth scene and press command and D to duplicate the scene.

Figure 3.162: Duplicate the scene.

3.6.71 Rename the scene to birdDance.

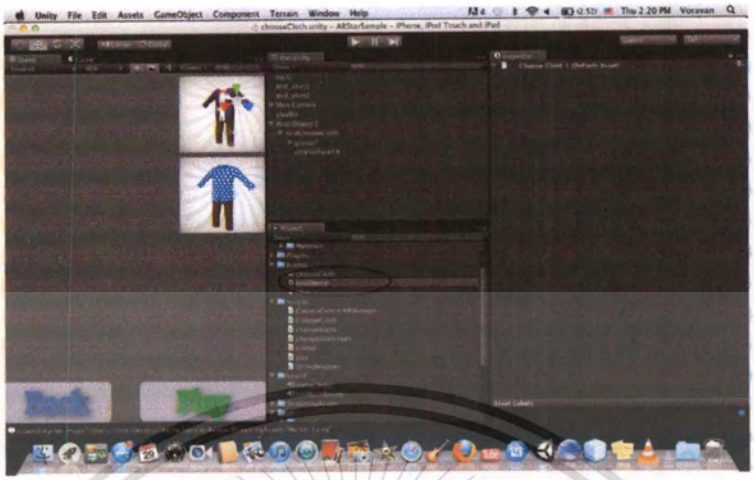


Figure 3.163: Rename the scene.

3.6.72 Open the birdDance scene. Drag and drop birdTooMuchDance.fbx to Root Object 1 in the Heirarchy view.

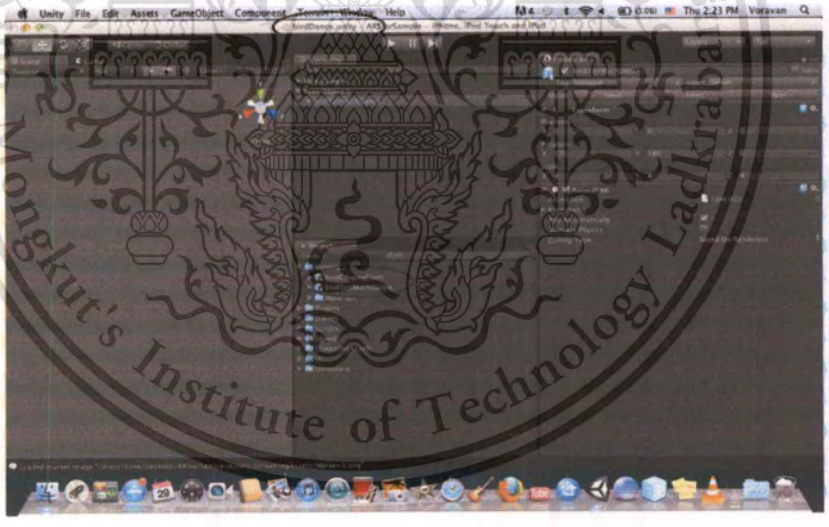


Figure 3.164: Drag and drop the model to Root Object 1.

3.6.73 Create mainBtn.png texture to GUI texture in the scene.

Figure 3.165: Create main button.

3.6.74 Arrange the position of Main button.

Figure 3.166: Arrange the position of the button.

3.6.75 Drag and drop changeScene.js script to mainBtn, which is in the Hierarchy view.

Figure 3.167: Drag and drop the script to main button.

3.6.76 Drag and drop mainBtn.png texture from UI folder in Project view to Normal Texture field and Roll Over Texture field in the Inspector view. Menu_beep.aiff audio clip from Sound folder in the Project view to Beep field in the Inspector view. Insert Main to Level to Load.

Figure 3.168: Drag and drop texture, audio clip to field in the Inspector view.

3.6.77 Select pause.png texture and create the GUI texture.

Figure 3.169: Create GUI texture.

3.6.78 Arrange the pause button position.

Figure 3.170: Arrange the pause button position.

3.6.79 Create the Java Script file. Rename it to pause.

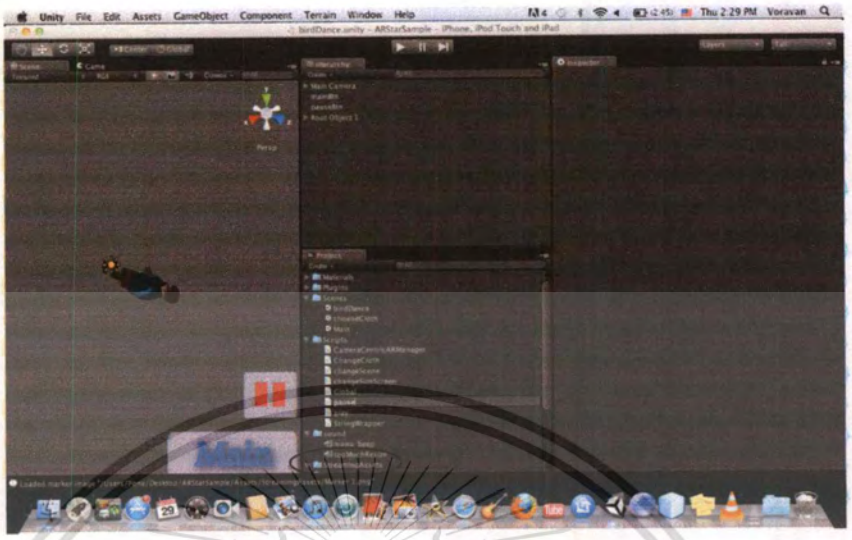


Figure 3.171: Create java script file.

3.6.80 Code the pause script.

```
private var isPause : boolean = false;

function OnMouseUp ()
{
    if(isPause) //play
    {
        Time.timeScale = 1;
        isPause = false;
        GameObject.Find("birdTooMuchDance").audio.Play();
    }
```

```
        else //pause
        {
            Time.timeScale = 0;
            isPause = true;
            GameObject.Find("birdTooMuchDance").audio.Pause();
        }
}
function Update()
{
    for (var touch : Touch in Input.touches)
    {
        if(guiTexture.HitTest(touch.position) && touch.phase ==
TouchPhase.Ended )
        {
            if(isPause)
            {
                Time.timeScale = 1;
                isPause = false;
```

```
GameObject.Find("birdTooMuchDance").audio.Play();
                }
                else
                {
                        Time.timeScale = 0;
                        isPause = true;
GameObject.Find("birdTooMuchDance").audio.Pause();
                }
        }
}
```

3.6.81 Drag and drop pause.js script to pauseBtn in the Hierarchy view.

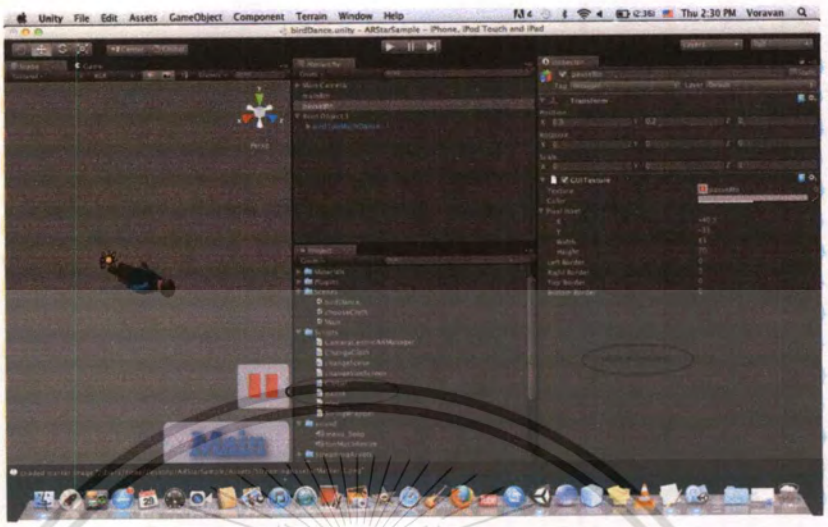


Figure 3.172: Drag and drop the script to Hierarchy view.

3.6.82 Create the dance script.

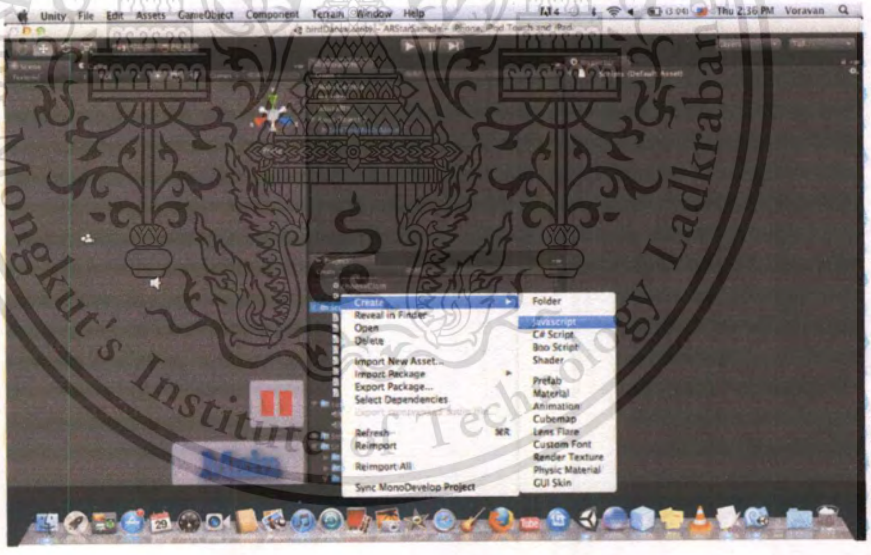


Figure 3.173: Create java script file.

3.6.83 Write the code for dancing.

```
var target : GameObject; // target : put the old texture inside, which is
choosen by player

// call recorded action

function Start()

{

        target.renderer.material.mainTexture = Global.currentCloth;

}

function Update () {

}
```

3.6.84 Drag and drop dance.js script to the birdTooMuchDance.

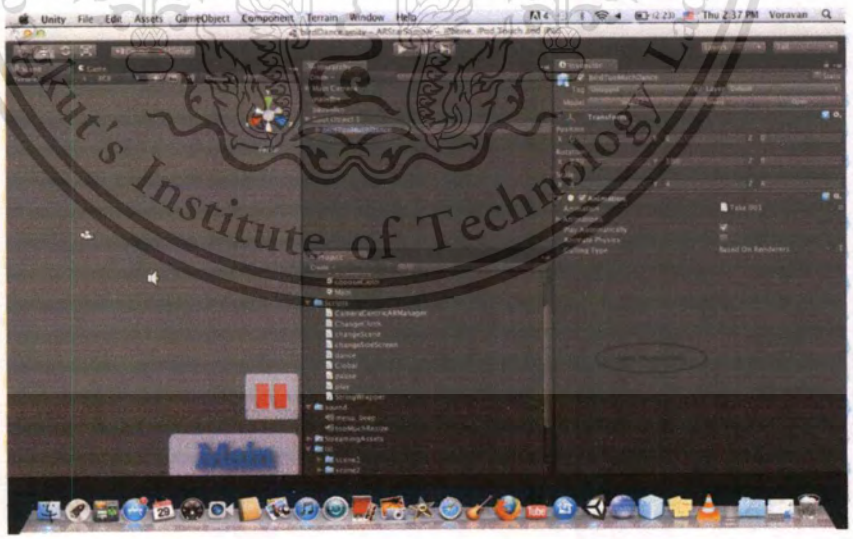


Figure 3.174: Drag and drop the script to the model.

3.6.85 Drag and drop the UV object to the Target field in the Inspector view.

Figure 3.175: Drag and drop the script to Hierarchy view.

3.6.86 Create the song by select Component>Audio>Audio Source.

Figure 3.176: Create Audio source.

3.6.87 Drag and drop tooMuchResize.mp3 audio clip to Audio Clip field in the Inspector view.

Figure 3.177: Drag and drop the audio clip to Inspector view.

3.6.88 Create the particle effect by import the package. Select Assets>Import Package>Particles.

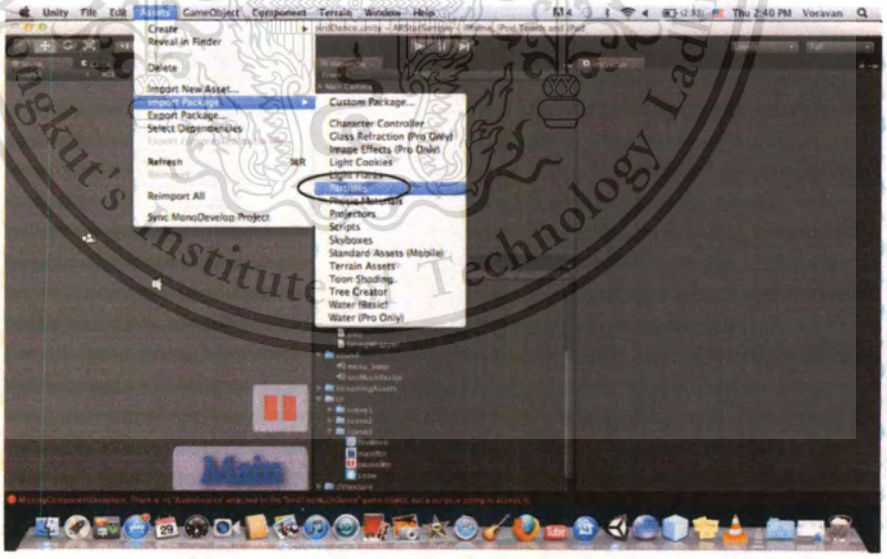


Figure 3.178: Create the particle effect.

3.6.89 Select all.

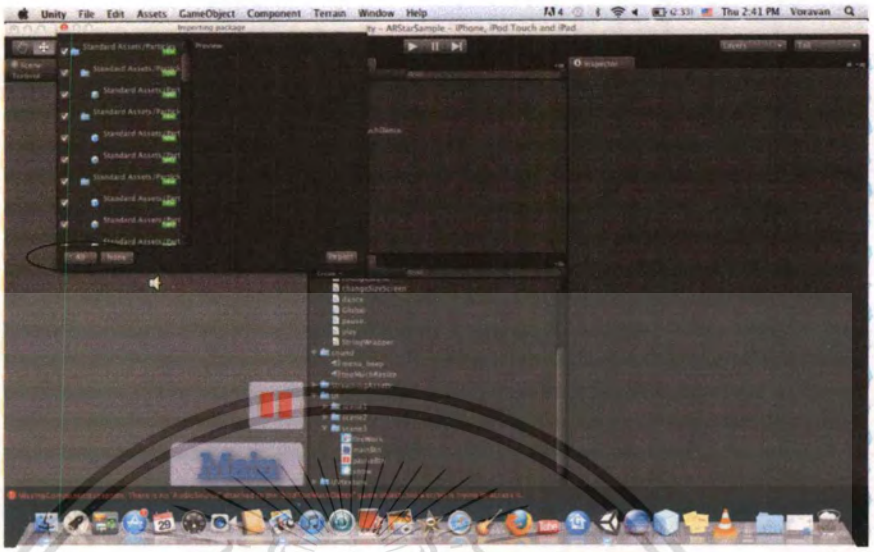


Figure 3.179: Select all.

3.6.90 Press Import.

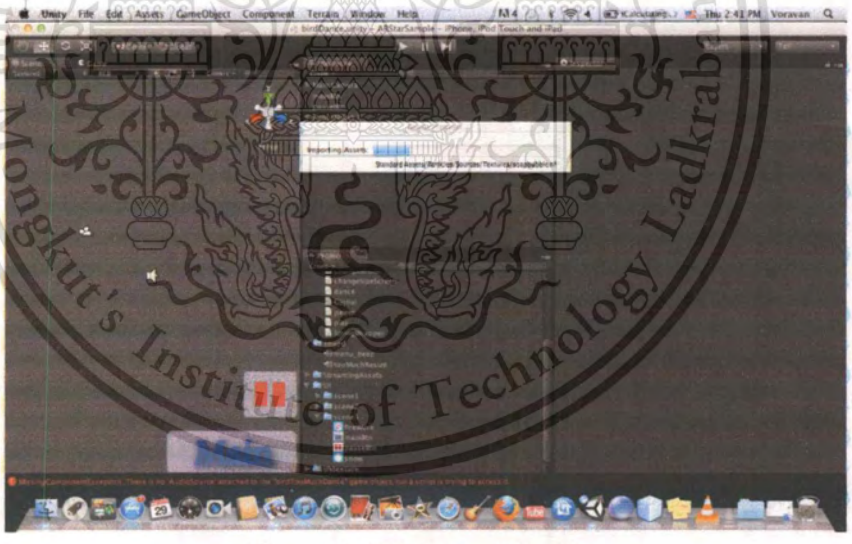


Figure 3.180: Import the file.

3.6.91 The package will provide the Standard Assets folder in the Project view.

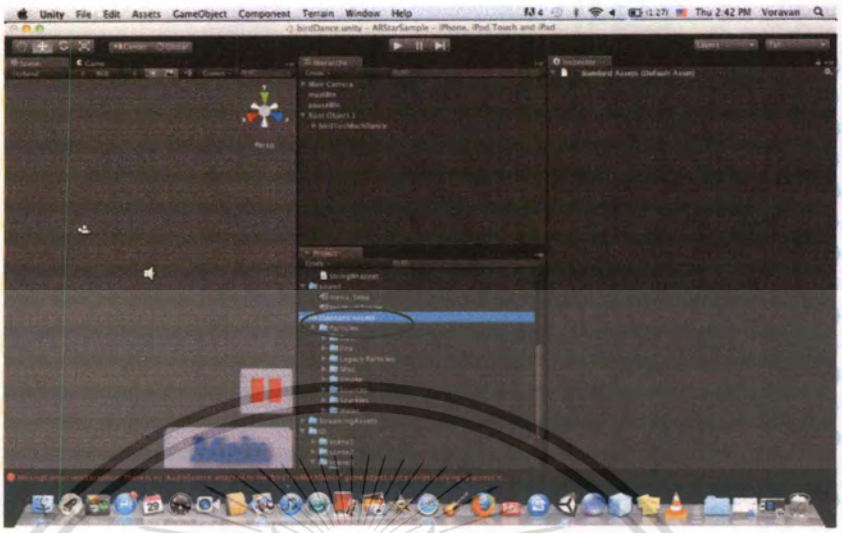


Figure 3.181: Standard Assets folder.

3.6.92 Create new folder and rename to Prefab.

Figure 3.182: Create Prefab folder.

3.6.93 Right click at the Prefab folder and select create prefab.

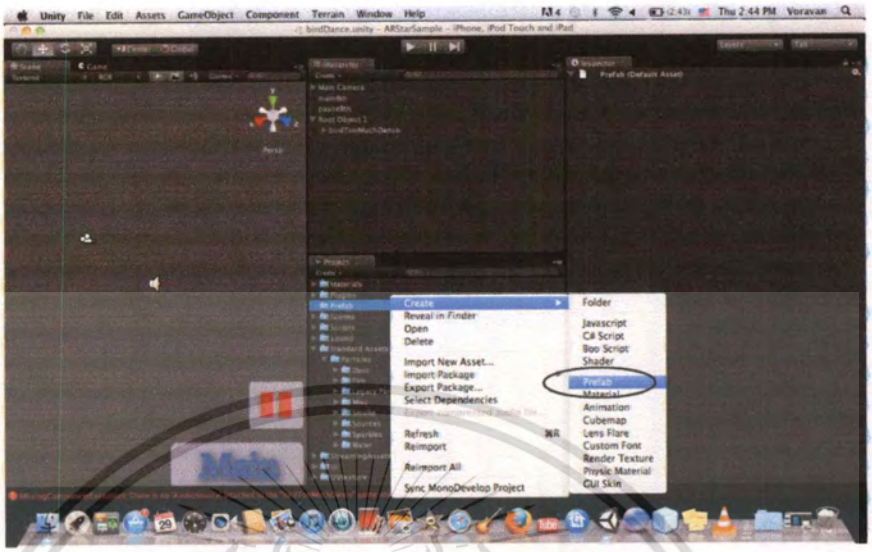


Figure 3.183: Create prefab.

3.6.94 Create two of prefab name firework and snow.

Figure 3.184: Rename new prefab.

3.6.95 Drag and drop Fireworks file in the Standard assets to the firework file in Prefab folder.

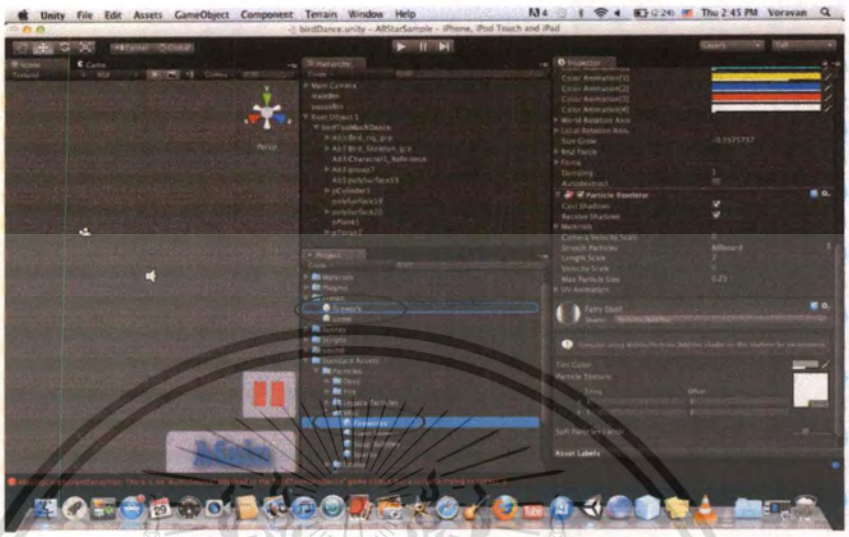


Figure 3.185: Drag and drop particle effect to prefab.

3.6.96 Drag and drop Light Snow file in the Standard assets to the snow file in Prefab folder.

Figure 3.186: Drag and drop particle effect to prefab.

3.6.97 Create new Java Script file and rename the file to createEffect.

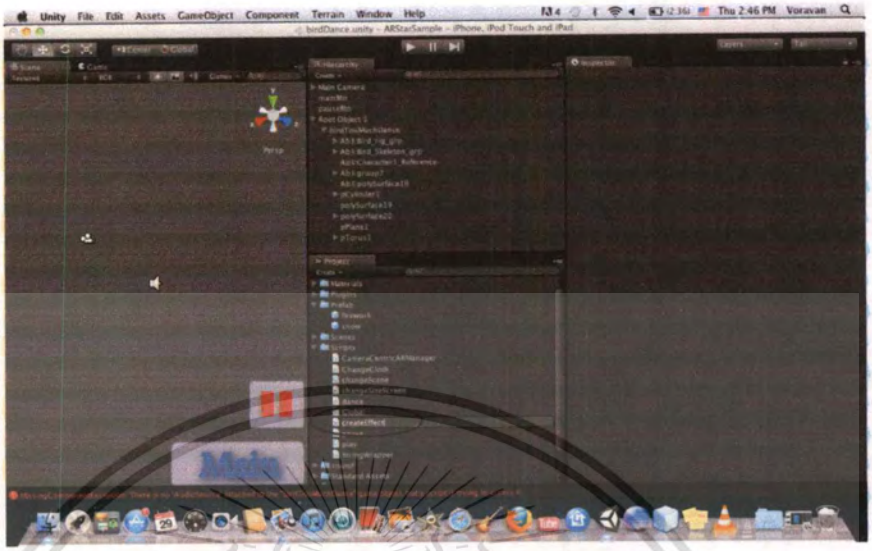


Figure 3.187: Create the Java script.

3.6.98 Write the script to create effect.

```
var particle : GameObject;

function Update()

{

        for (var touch : Touch in Input.touches)

        {

                if(guiTexture.HitTest(touch.position) && touch.phase ==

                        TouchPhase.Stationary)

                {
```

```
Instantiate(particle,Vector3(0,0,0),Quaternion.identity);
            }


        }

        if(Input.GetKey(KeyCode.LeftArrow))
        {
            Instantiate(particle,Vector3(0,0,0),Quaternion.identity);
        }
}
```

3.6.99 Select firework.png texture, which is in the UI folder in the Project view.

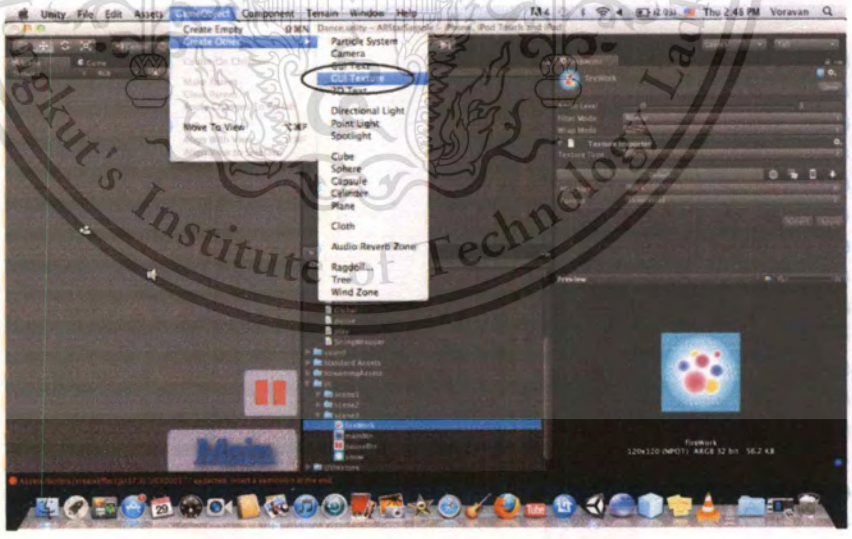


Figure 3.188: Create GUI texture.

3.6.100 Drag and drop createEffect.js script to the firework button in the Hierarchy view.

Figure 3.189: Import the script to button texture.

3.6.101 Drag and drop the firework prefab to the Particle field in the Inspector view.

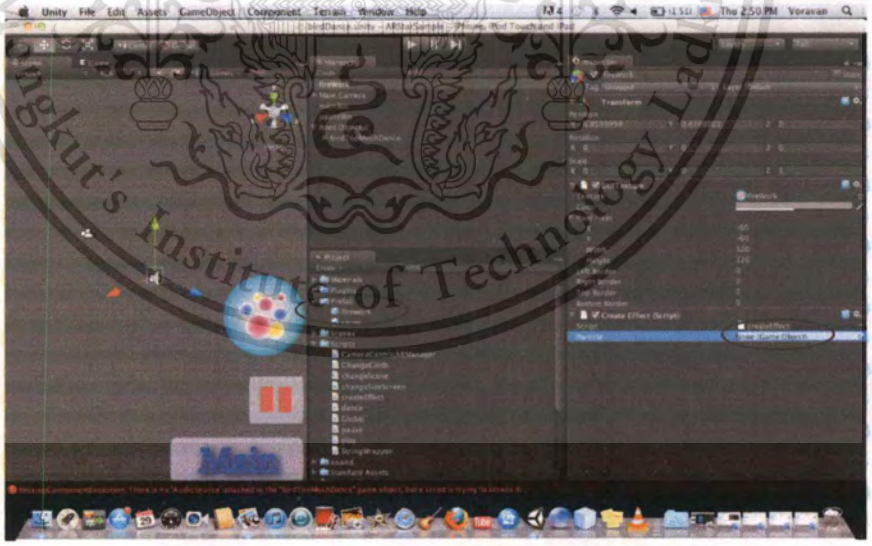


Figure 3.190: Import the prefab to button texture.

3.6.102 Create the snow effect button by repeat previous step. Drag and drop the snow prefab to the Particle field in the Inspector view.

Figure 3.191: Snow effect button.

3.6.103 Create Java Script file for create hide script. When user touch on the screen, all the buttons will appear and after 5 second, all the buttons will disappear.

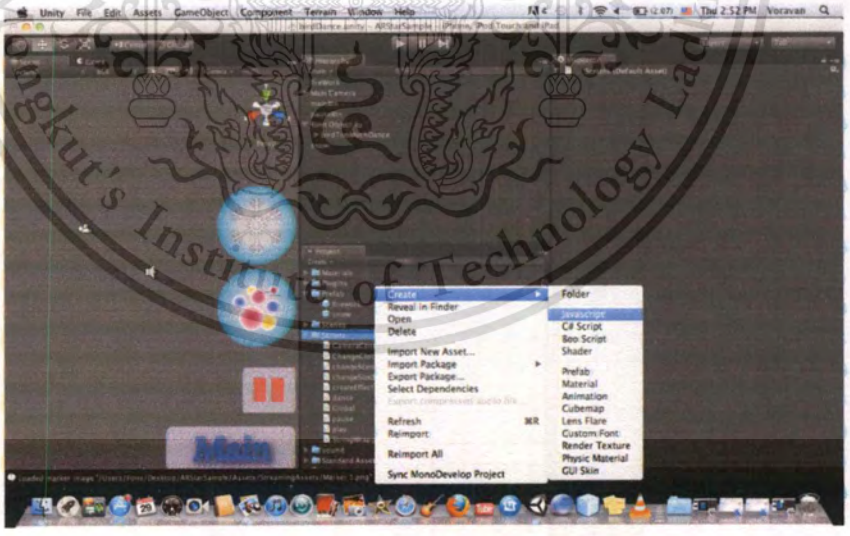


Figure 3.192: Create Java Script file.

3.6.104 Rename the script to hideButton.

Figure 3.193: Rename the script.

3.6.105Write the hide script by using Java Language.

```
var allButton : GUITexture[];

private var showing : boolean;

private var timer : float = 0.0f;

function Start()

{

        setButton(false);

}
```

```
function Update () {

    for (var touch : Touch in Input.touches)

    {

        if(touch.phase == TouchPhase.Began)

        {

            setButton(true);

        }

    }

    if(Input.GetMouseButtonUp(0))

    {

        setButton(true);

    }

    if(showing)

    {

        timer+=Time.deltaTime;
```

```
if(timer>5){
                    setButton(false);
                    timer=0;
                }
            }
}

function setButton(val:boolean){
        for(var i=0;i<allButton.length;i++)
        {
                allButton[i].enabled=val;
        }
        showing=val;
}
```

3.6.106 Select GameObject>Create Empty.

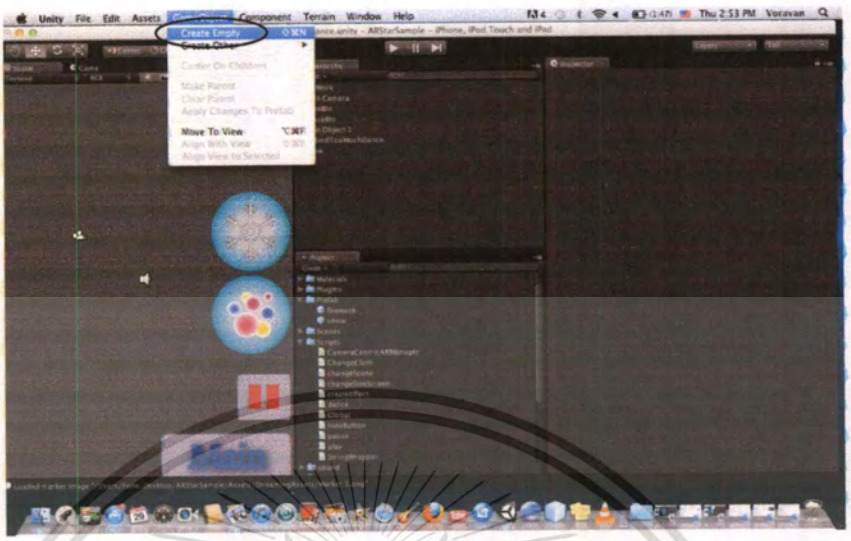


Figure 3.194: Create empty.

3.6.107 Rename it to Controller.

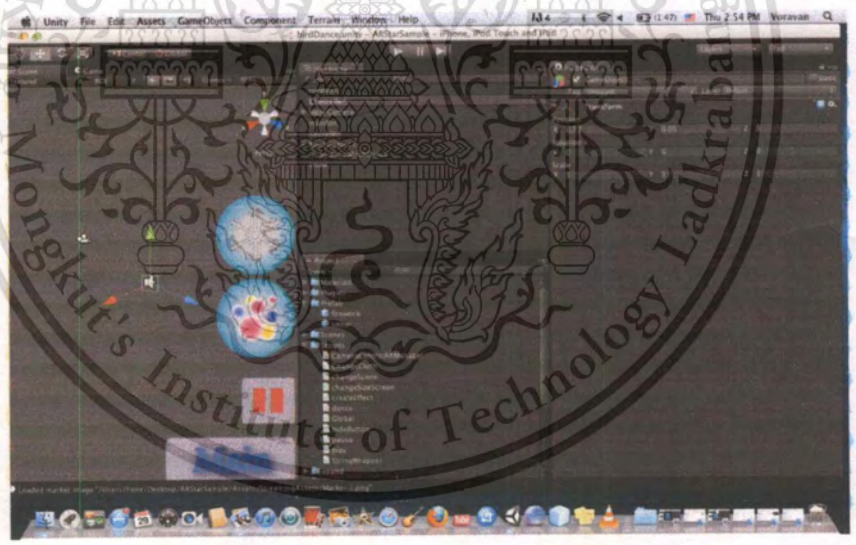


Figure 3.195: Rename to Controller.

3.6.108 Drag and drop hideButton.js from Scripts folder in Project view into empty space of Inspector view.

Figure 3.196: Drag and drop the script to Inspector view.

3.6.109 Expand all the Button and insert 4 to the Size field in the Inspector view.

Figure 3.197: Insert the number of hide button.

3.6.110 There are four elements appear on the Inspector view. Drag and drop all the buttons, which is in the Hierarchy view to each element.

Figure 3.198: Drag and drop the button to each field.

3.6.111 All the elements will linked to all the buttons.

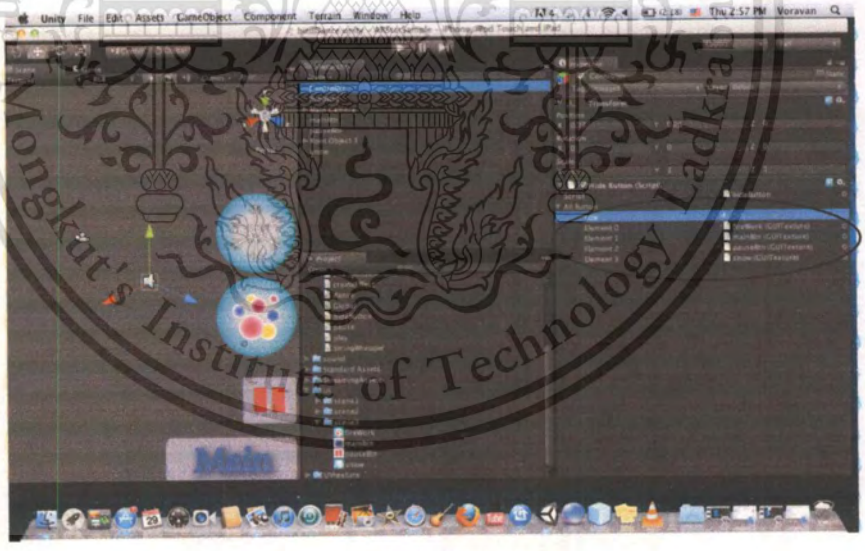


Figure 3.199: Hide button.

3.6.112Create new Java Script file.

Figure 3.200: Create Java script file.

3.6.113 Rename the script to Destroy. The destroy.js is used for setting the time of particle effect, which represent on the screen.

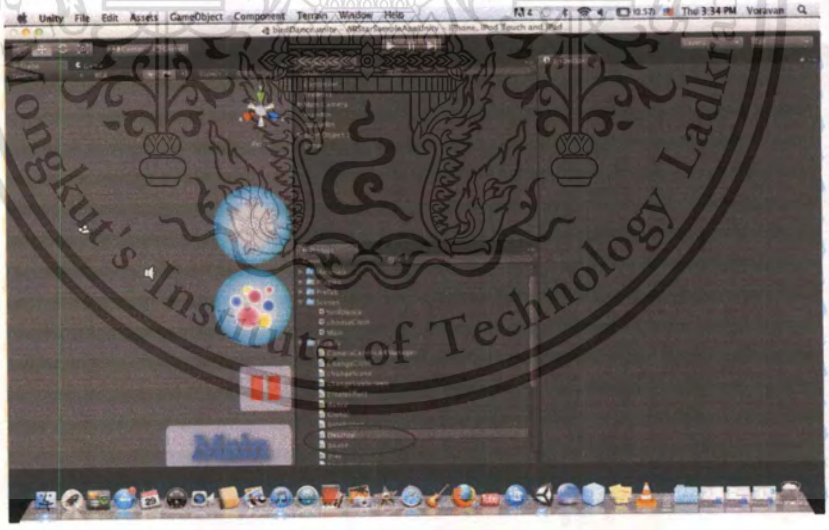


Figure 3.201: Rename the script.

3.6.114Write the destroy script by using java script language.

```
var time : float=5.0;

function Start() {

        Destroy(gameObject,time);

}
```

3.6.115 Drag and drop Destroy.js to firework in folder Prefab.

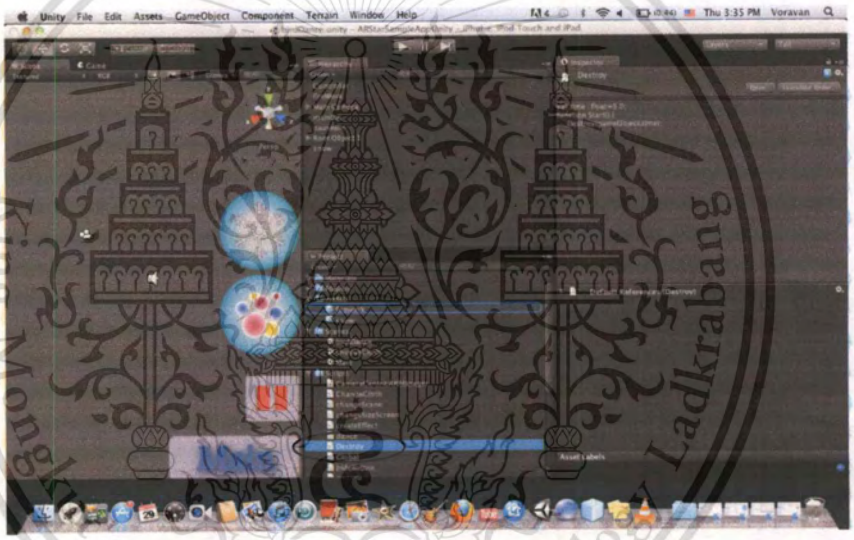


Figure 3.202: Drag and drop the script to prefab.

3.6.116 Drag and drop Destroy.js to snow in folder Prefab.

Figure 3.203: Drag and drop the script to prefab.

3.6.117 Select File>Build Settings to build the application.

Figure 3.204: Build the application.

3.6.118 Drag and drop all scenes to Scenes In Build space. Make sure that the first scene is on top.

Figure 3.205: Drag and drop all scenes to Scenes In Build space.

3.6.119 Press the Build button. Save in ARStarSample name, and press Save.

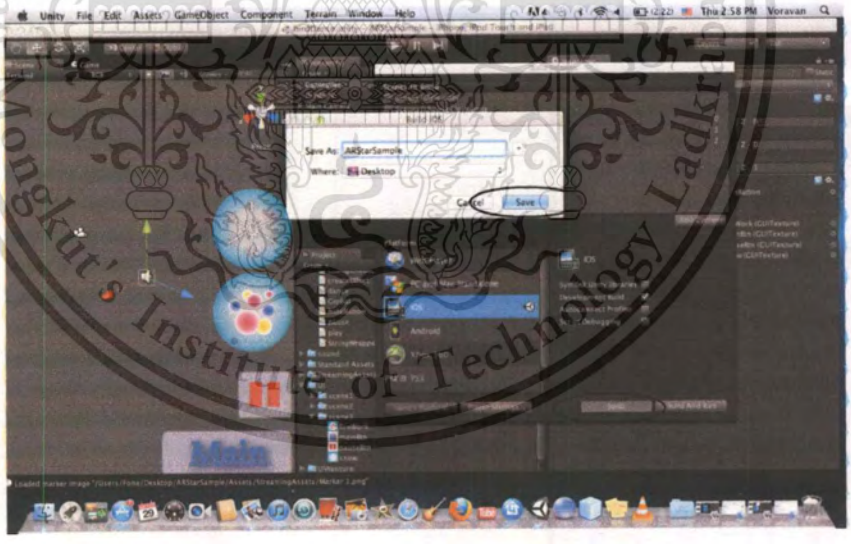


Figure 3.206: Press the Build button.

3.6.120 Build processing. The XCode file will be on the Desktop.

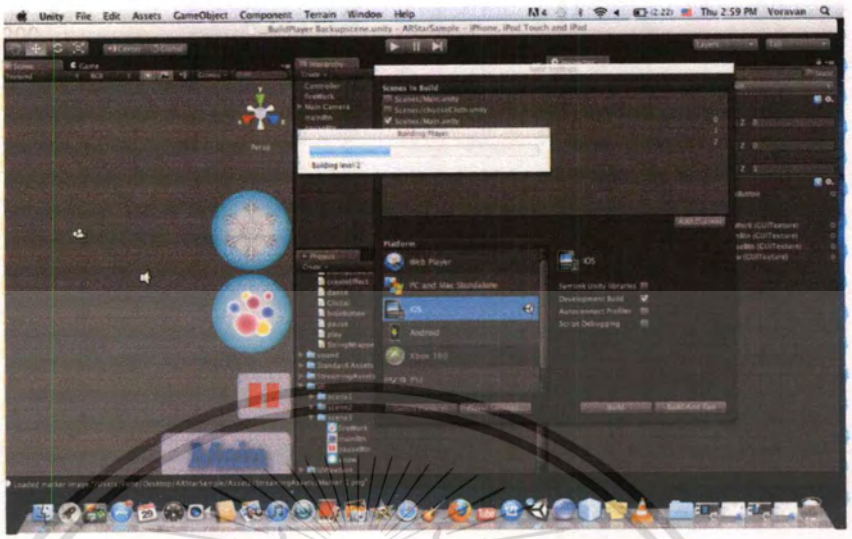


Figure 3.207: Build processing.

Chapter 4

Implementation

4.1 Install the application

4.1.1 Building for an iOS device

4.1.1.1 Select Edit → Project Settings → Player.

In the Inspector view, select Other Settings. Set Bundle Identifier to com.yourcompany.YourApp. Set Target Device to iPhone + iPad. Set Target Platform to Universal armv6+armv7. Set SDK Version to iPhone iOS latest.

Figure 4.1: Build Setting.

4.1.1.2 Choose iOS format and press the Build button.

Figure 4.2: Choose iOS format and press the Build button.

4.1.1.3 Choose a folder for your build and press Save.

Figure 4.3: Choose the folder for save file.

4.1.1.4 Wait until it built successful.

Figure 4.4: Build running.

4.1.1.5 The file will be on the desktop, open Unity-iphone.xcodeproj file.

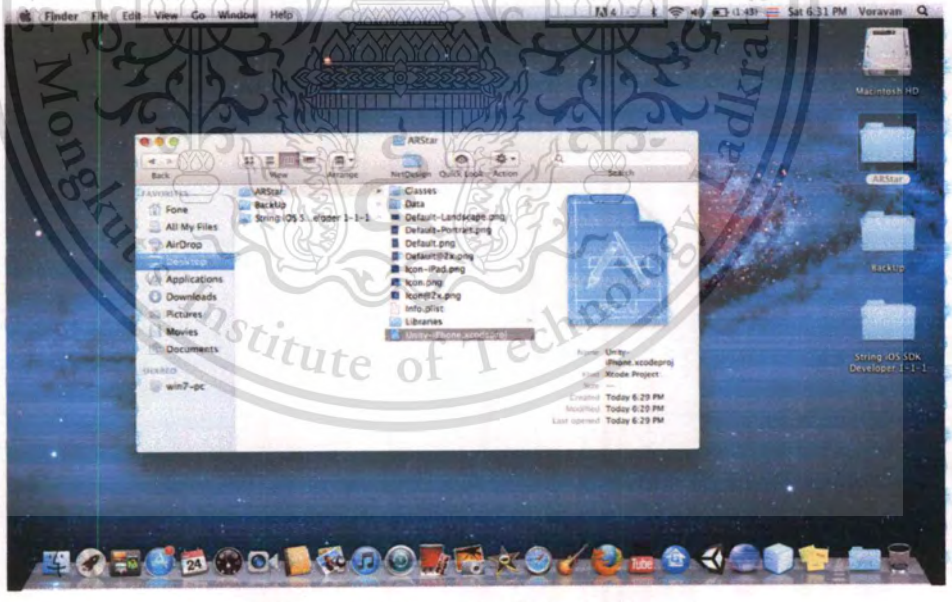


Figure 4.5: Open the XCode file.

4.1.1.6 In Xcode, stop the build. In the Project navigator, select Unity-iPhone → target Unity-iPhone → Build Phases tab, and expand Link Binary With Libraries. Click the plus button, then press and hold cmd and select the AVFoundation framework and click Add.

Figure 4.6: Set up the library in XCode.

4.1.1.7 Repeat on 4.1.1.6 and add CoreGraphics framework.

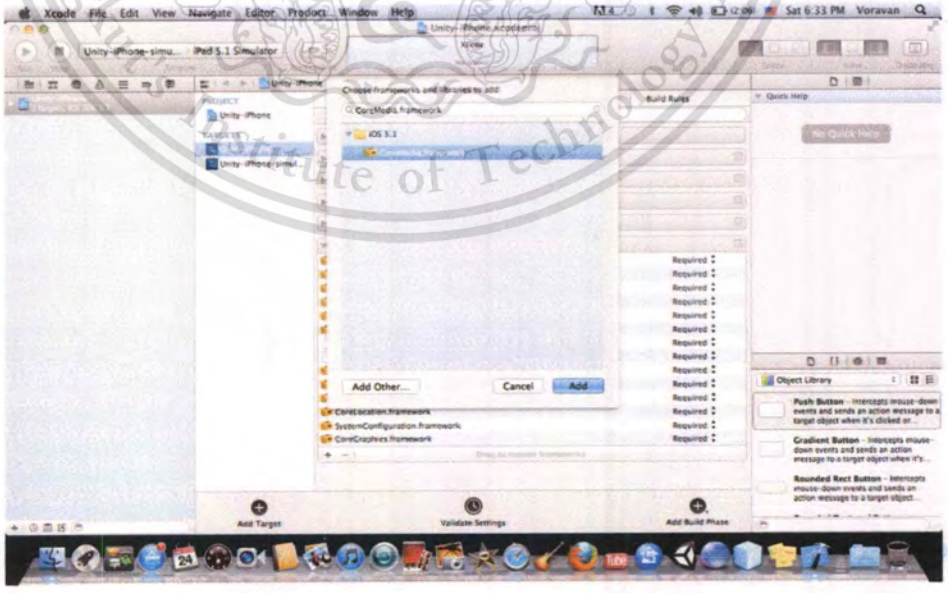


Figure 4.7: Add CoreGraphics framework.

4.1.1.8 Repeat on 4.1.1.6 and add CoreMedia framework.

Figure 4.8: Add CoreMedia framework.

4.1.1.9 Repeat on 4.1.1.6 and add CoreVideo framework.

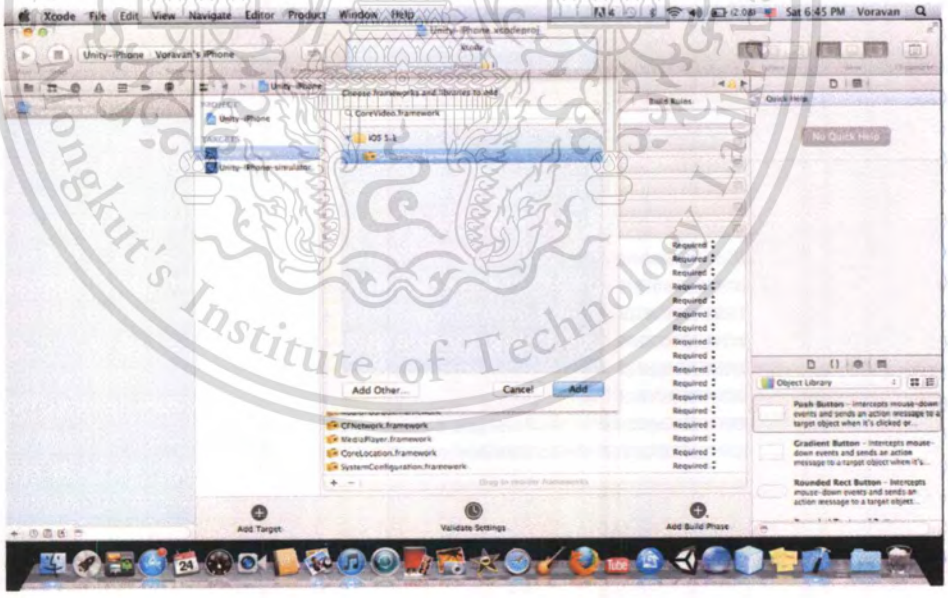


Figure 4.9: Add CoreVideo framework.

4.1.1.10 Repeat on 4.1.1.6 and click Add Other.

Figure 4.10: Add Other.

4.1.1.11 Navigate to the unzipped String SDK and select libStringUnity*.a from the Libraries folder. Click Add.

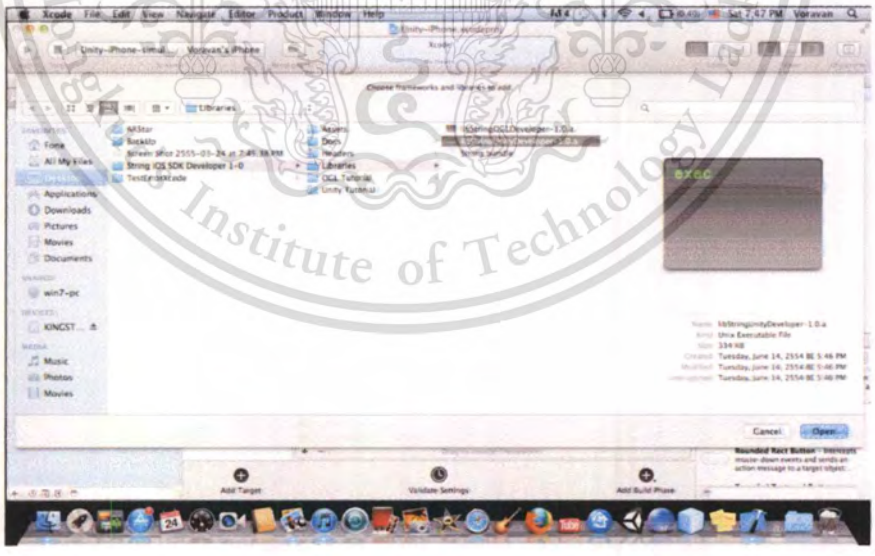


Figure 4.11: Select libStringUnity*.a from the Libraries folder.

4.1.1.11 In the Project navigator, select Unity-iPhone → project Unity-iPhone → Build Settings. Make sure that the viewing All and not Basic mode. In the search field, type Preprocessor Macros.

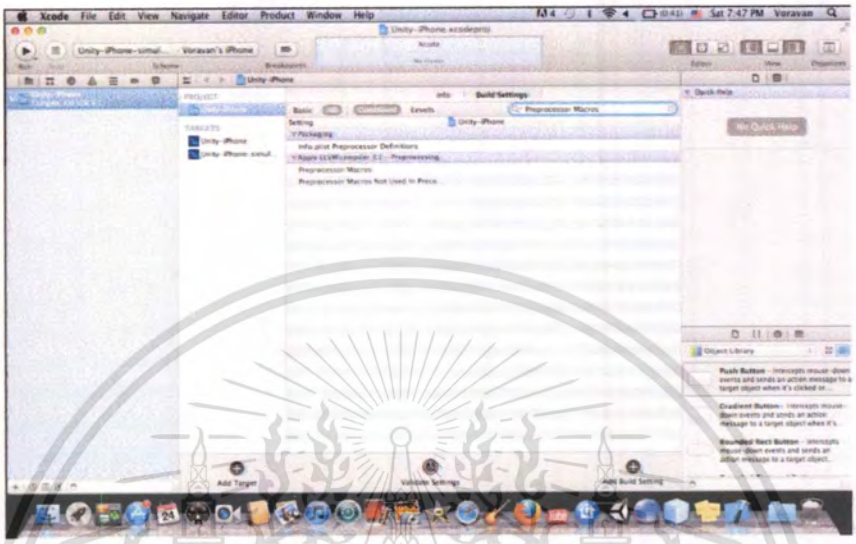


Figure 4.12: Find Preprocessor Macros from search field.

4.1.1.12 In the Preprocessor Macros field, copy and paste the following:

PresentContext_UnityCallback(x)=PresentContext_UnityCallback_Old(x)

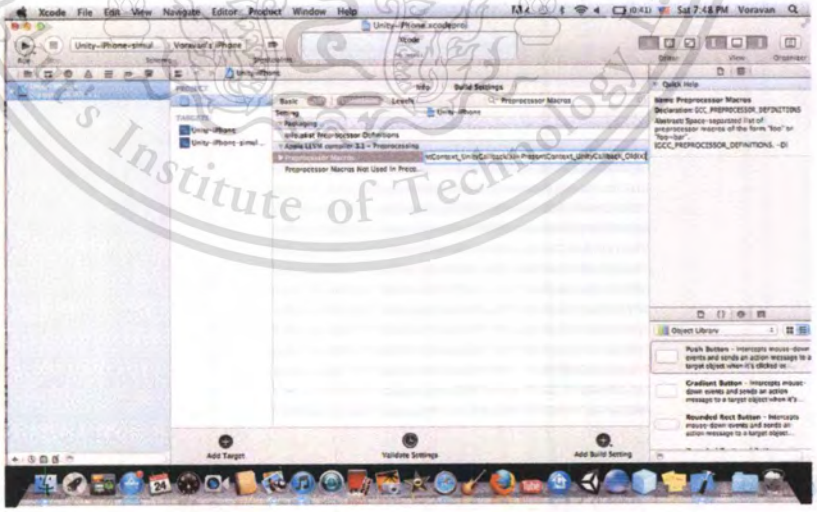


Figure 4.13: Paste the script in the Preprocessor Macros field.

4.1.1.13 Make sure the active scheme is iOS Device. Press cmd+B to build. Attach an iOS device with a camera that enabled for development. Press cmd+R to run. After the app has loaded, point the IPhone camera at one of the printed markers.

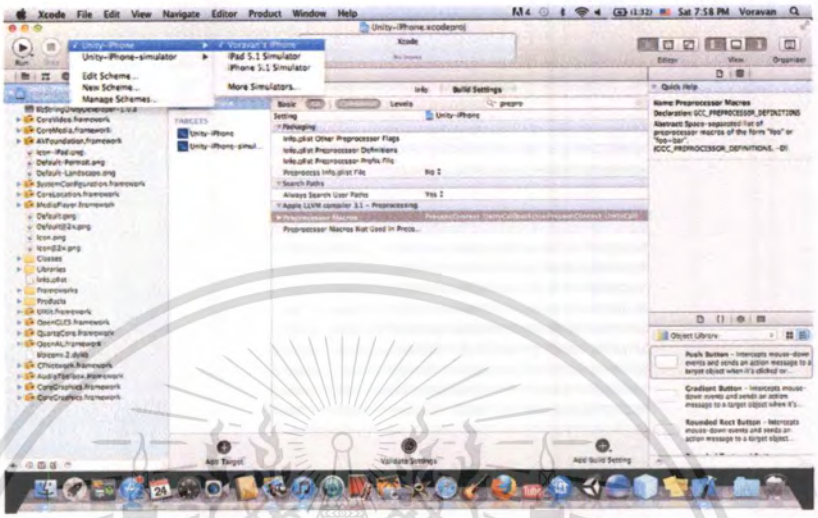


Figure 4.14: Build the file to IPhone.

4.2 The explanation of the User Interface

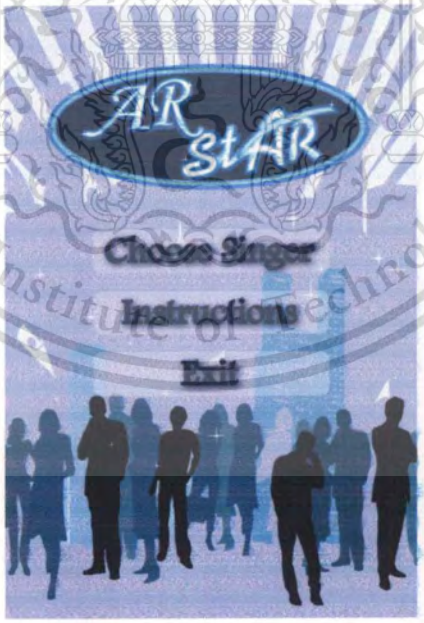


Figure 4.15: Main screen.

The index shows the mainly functions of the application including Choose Singer, Instructions and Exit

4.2.1 Choose Singer

The user has to touch on Choose Singer button to choose the singers that they want.

4.2.2 Instructions

This button gives user an instruction of this application.

4.2.3 Exit

The user has to click the Exit button to quit the application.

Figure 4.16: Instruction screen.

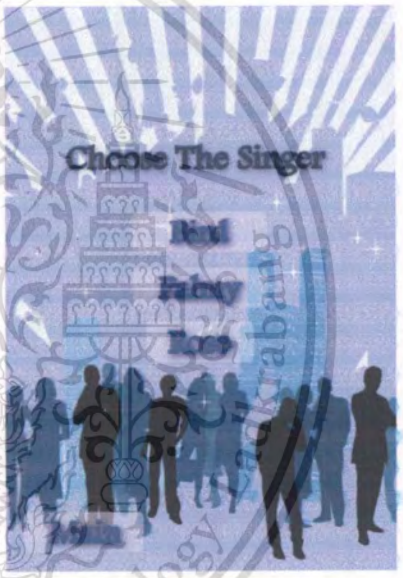


Figure 4.17: Choose The Singer screen.

4.3 How to play

4.3.1 After the user touch on "Choose Singers" button, the application will show "Choose The Singer" screen. The user can choose the singer they want.

There are three different markers, which match to each singer.

Figure 4.18: Bird marker and the virtual model.

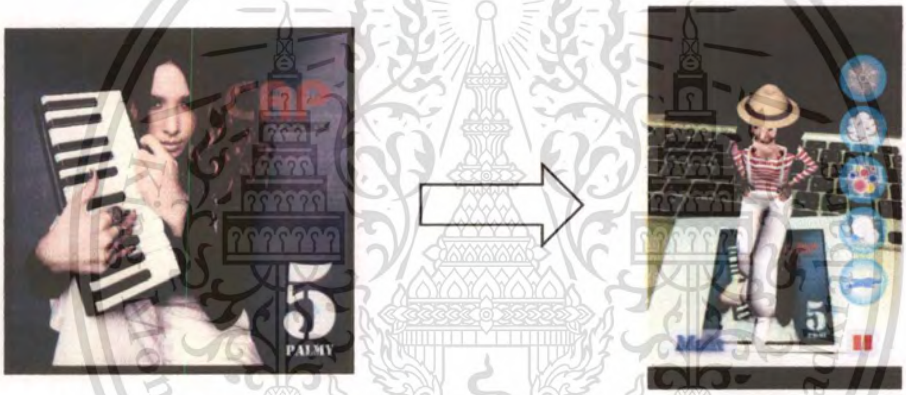


Figure 4.19: Palmy marker and the virtual model.

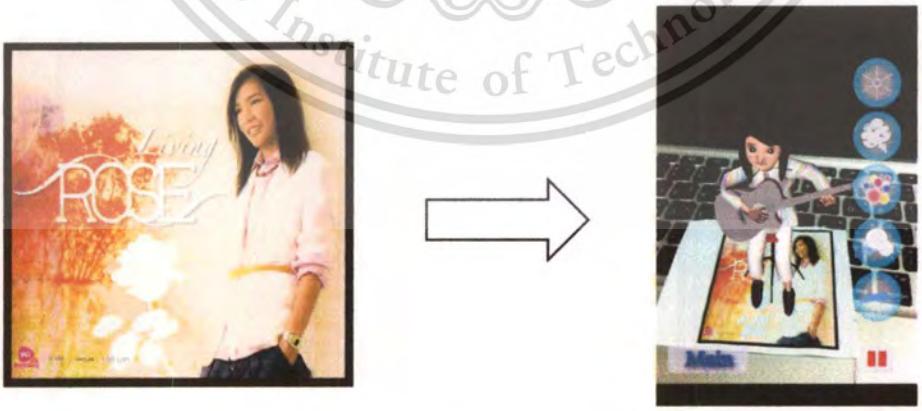


Figure 4.20: Rose marker and the virtual model.

4.3.2 After the user choose the singer, they can choose the song.

Figure 4.21: Choose The Song screen.

4.3.3 The user can enjoy with alternative choice of singer cloths.

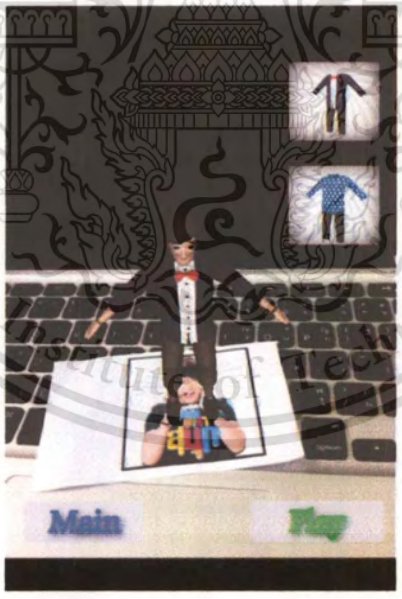


Figure 4.22: Choose The Cloth screen.

4.3.4 After the user touch Play button, the application will show the dance screen. The user can pause the song, back to main screen and choose the particle effect including the snowflake, ice smoke, firework, sparkle, and light surface.

Figure 4.23: Particle effect.

Chapter 5
Conclusion

5.1 Conclusion

This special project has achieved into a new innovation between AR application and the music for the IPhone. The users will have alternative choice to enjoy visual and audio simultaneous. They can interact with the application by choosing the singers, songs, and cloths. Also particle effects including snowflake, smoke, firework, sparkle, and light. When the user plays a song, they can pause and play. The application can simulate the singer's 3D model into the real time while it moves the marker out from the IPhone camera.

String™ Augmented Reality used to receive parameters from IPhone camera, analyzes the marker, and calculates from the marker position and represents the 3D model on the marker through the IPhone screen. Unity is used for manage about create the application, sequence in the scene in application, control the singer model. The source code for control action was created by using Java Script language, including the action when player touch the button, singer model change the cloth, the particle effect in the dance scene. Xcode provide the framework for connect the application to IPhone including media framework, graphic framework, and video framework.

The source code for controlling the action was created by using Java Script language, including the action when player touches the button to have singer model changing the cloth, and the particle effect in the dance scene. The Xcode provides the framework for connecting the application to IPhone including media framework, graphic framework, and video framework.

5.2 Limitation

After researched and developed the application with Augmented Reality technology, the problems have occurred as follows:

5.2.1 If there are dim light sources, the application will be more difficult to track the marker, and the singer model will disappear

5.2.2 The singer model take more memory to create a large size of animation. During the user is pressing the Play button, it will take more time to process the dance scene.

5.2.3 During the singer model is dancing, and if the IPhone or the marker is unstable, this will interrupt dancing process and it leads to an application error.

5.3 Benefit

5.3.1 Benefit to the system developers

5.3.1.1 Benefit from learning and gaining experience from developing AR technology this becomes remarkable.

5.3.1.2 Benefit from studying how to make application for Iphone.

5.3.1.3 Benefit from learning how to make 3D animation model by Autodesk Maya.

5.3.1.4 Benefit from studying how to manage the application in Unity.

5.3.2 Benefit to the users

5.3.2.1 User has alternative choice to choose entertainment technology in listening to the music.

5.3.2.2 In the business, the user can use this application to apply in their businesses, and make more profit to them.

5.3.2.3 Approach to further study of AR technology.

5.4 Suggestion

Even though the Music Video Augmented reality application for IPhone makes life more enjoyable for the user, but it has some limitation on the technology. During using the application, the user should be in the place where it has proper light. To collect a large memory's size on 3D model in the application, and it will not robustness. When create the polygon, it should create the low polygon model.

References

[1] http://turbulence.org/blog/2011/04/07/empyre-re-emergence-of-the-augment/[2] http://www.cs.unc.edu/Research/ProjectSummaries/ultrasou.pdf
[3] http://www.arlab.nl/old/docs/interactive_AR.pdf

[4] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.63.4105&rep=rep1&type=pdf[5] http://www.arlab.nl/old/docs/interactive_AR.pdf[6] http://www.ijcaonline.org/volume5/number5/pxc3871290.pdf

[7] http://www.ijcaonline.org/volume5/number5/pxc3871290.pdf[8] http://www.ubergizmo.com/2010/05/tissot-lets-you-try-its-t-touch- watches-via-augmented-reality/[9] http://www.hitl.washington.edu/artoolkit/documentation/cs.htm [10] http://www.hitl.washington.edu/artoolkit/documentation/cs.htm

Appendix A.

Autodesk Maya 2012 Installation.

Autodesk Maya 2012 Installation.

1.Double-click the Maya executable file and click next to unzip the extracted files to the default directory (C:\Autodesk\Maya2012). In the next welcome screen, choose Install. Wait until the installer has initialized.

Figure A1: Welcome screen of Maya 2012.

2. In the License Agreement window, do the following: Select Country or Region from the drop-down list. Select I ACCEPT to accept the terms and conditions. Click Next.

Figure A2: License Agreement window.

3. In the Product Information window, select **Stand-Alone** as License Type.

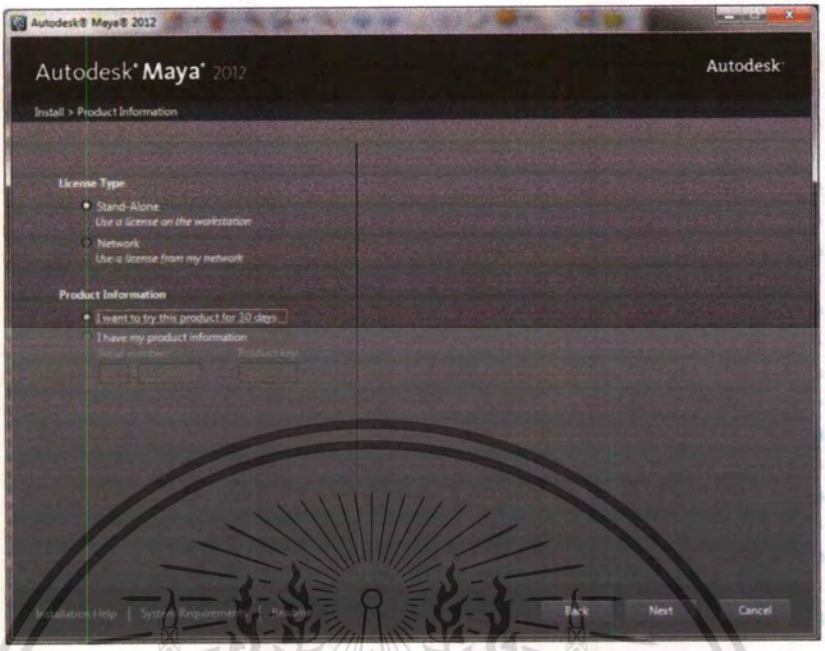


Figure A3: Product Information window.

4. In the Configure Installation window, select the components you want to install. Select Maya 2012 and Direct Connect 2012.

Figure A4: Configure Installation window.

5. An Installation Progress window displays. This may take a while.

Figure A5: Installation Progress window displays.

6. Verify that all components have been installed successfully. Click Finish.

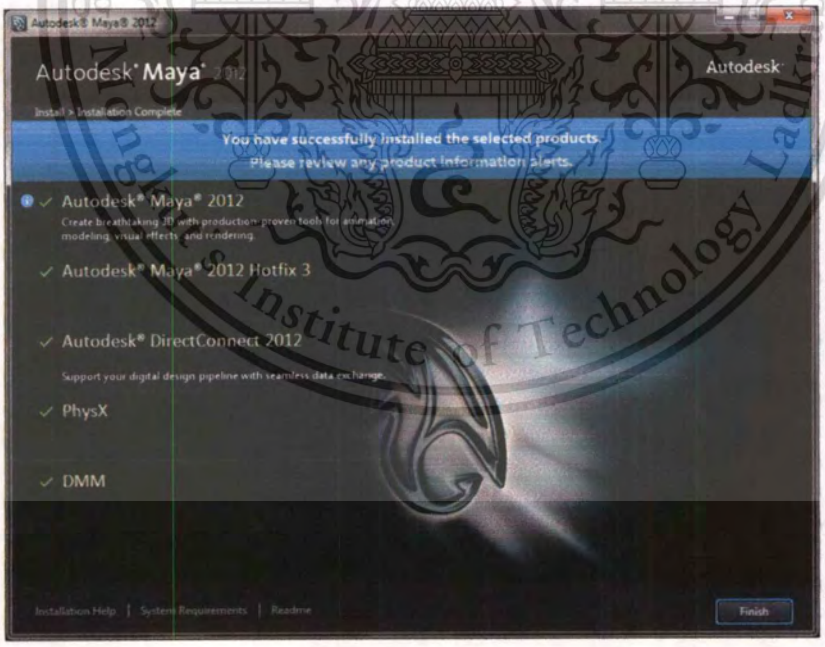


Figure A6: Installation complete screen.

Appendix B.

Adobe Photoshop CS5 Installation

Adobe Photoshop CS5 Installation

1.In the first screen, there is the Photoshop CS5 welcome installation. Click the Accept button to continue.

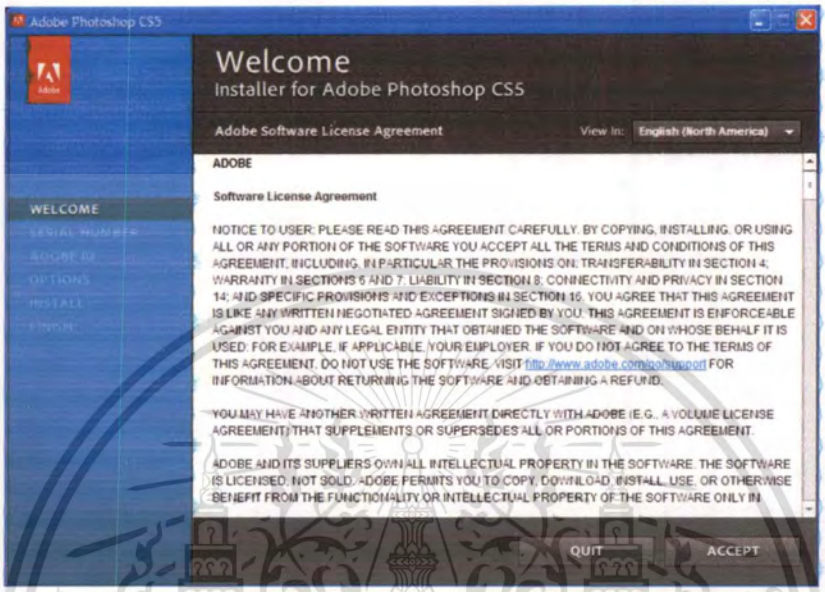


Figure B1: First screen of Photoshop CS5 installation.

2. In the second screen, type serial number of Photoshop CS5.

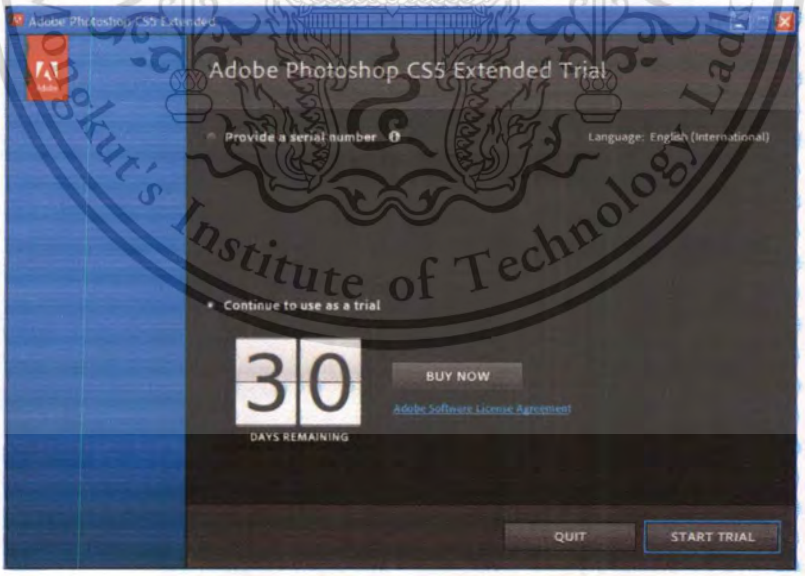


Figure B2: License screen of Photoshop.

3. Select the installation location, default is *C:\Program Files\Adobe*. Program need 1.5 GB free space for this installation.

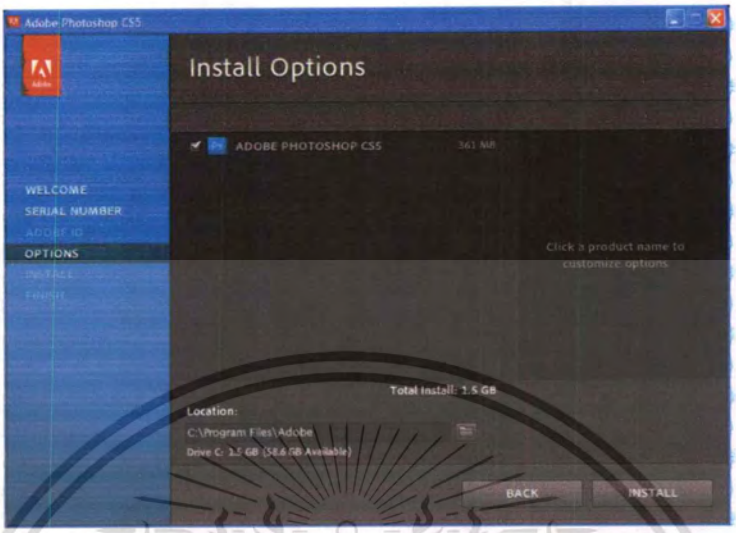


Figure B3: Install Option of Photoshop CS5

4. Wait for the installation software preparing Adobe Photoshop CS5 until finish.

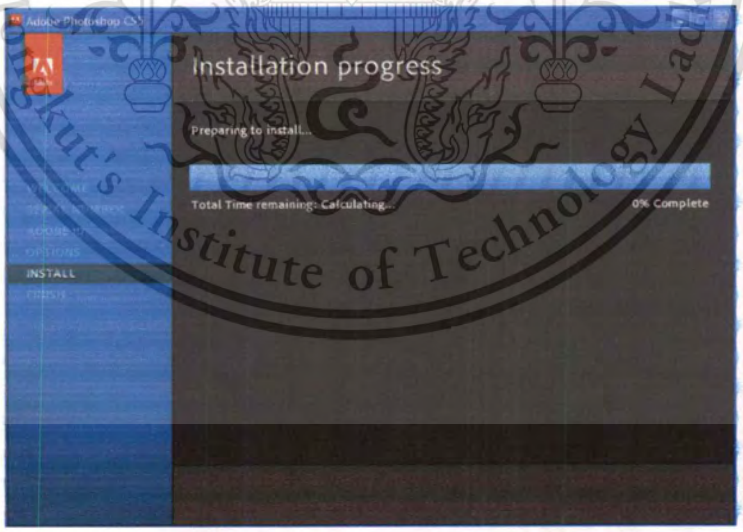


Figure B4: Installation progress screen.

5. After it all installed, users can click **Done** button. And Photoshop CS5 is installed.

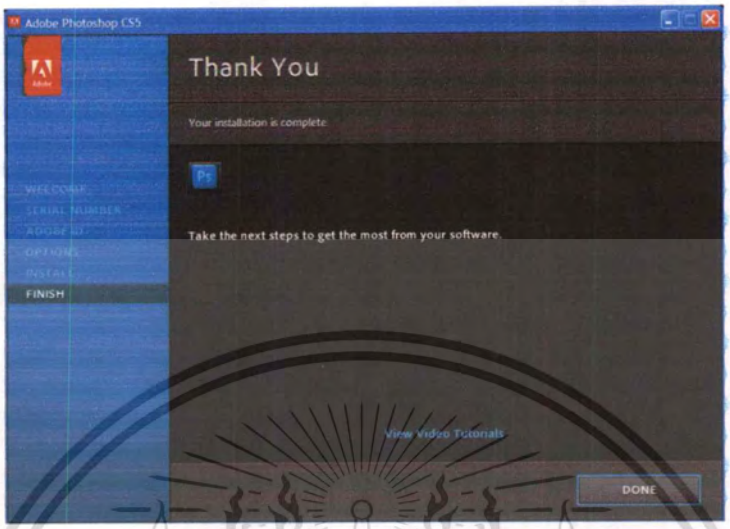


Figure B5: Installation Complete screen.

Appendix C.

Unity 3.4 Installation

Unity 3.4 Installation.

1.Double click to the Unity.dmg executetable file, and click to continue button.

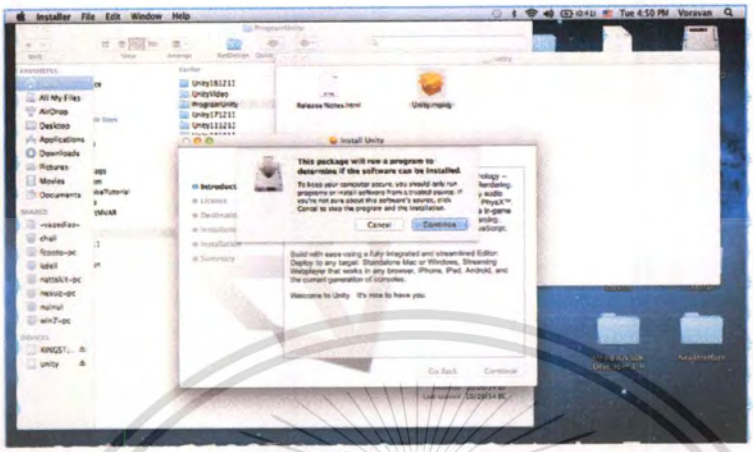


Figure C1: Unity.dmg executetable file.

2. In the next Unity install screen, click continue.

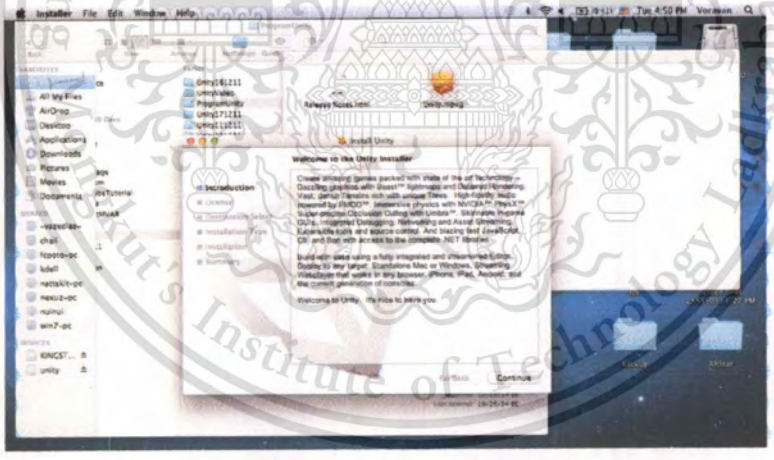


Figure C2: Unity install screen.

3. In the license screen click continue to next step.

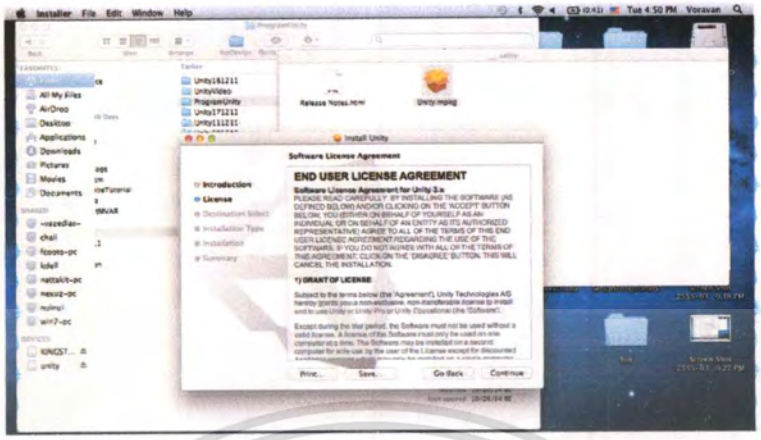


Figure C3: License screen of Unity.

4. Select the installation location. Program need 2.21GB free space for this installation, and click install.

Figure C4: Installation type screen of Unity.

5. Wait until the installer has initialized.

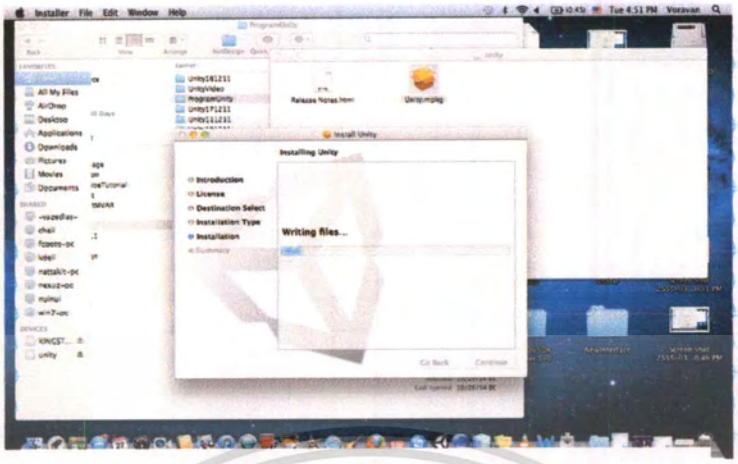


Figure C5: Screen of Unity installation.

Appendix D.

UVLayout Installation

UVLayout Installation

1. Run the supplied uvlayout.exe file.

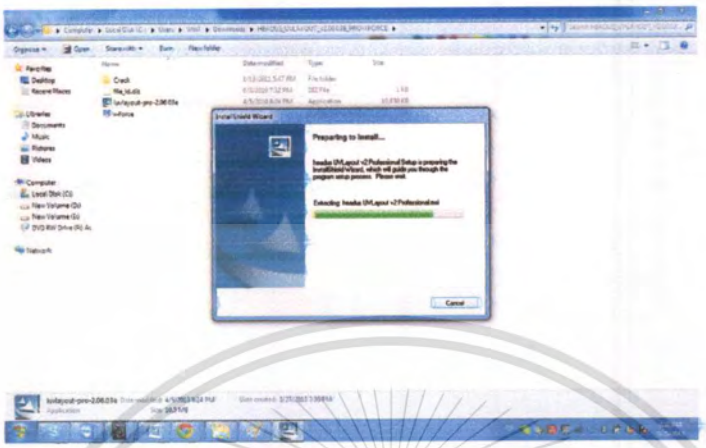


Figure D1:UVlayout Installation Wizard.

2. Follow the instructions displayed to install UVLayout.

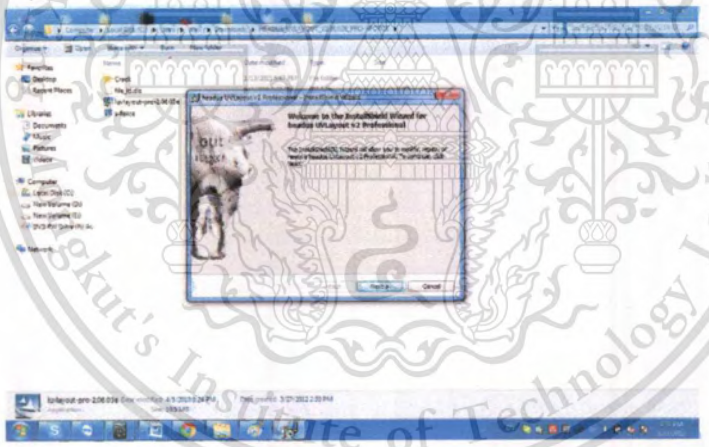


Figure D2: UVlayout installation screen.

3.When installation complete, activate program by license key into the keys file.

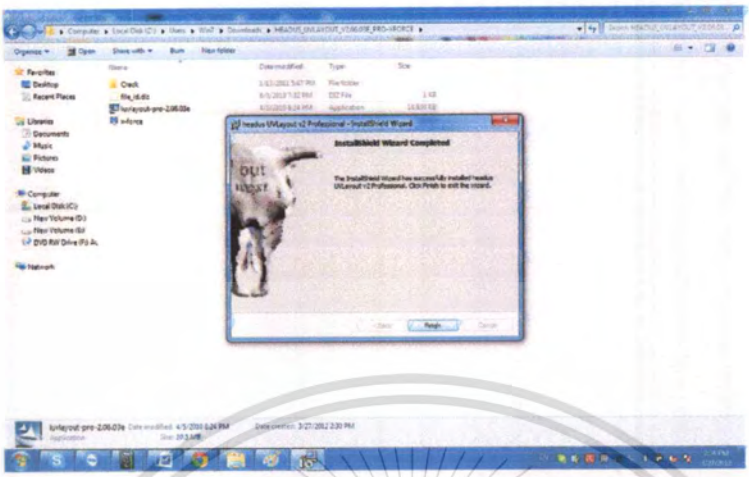


Figure D3: UVlayout installation complete screen.